

Informix Migration Guide

Informix Dynamic Server.2000, Version 9.2
INFORMIX-Universal Server, Version 9.10 to 9.14
Informix Extended Parallel Server, Version 8.3
Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, Version 8.21
INFORMIX-OnLine XPS, Version 8.10 to 8.11
Informix Dynamic Server, Version 7.30 to 7.31
Informix Dynamic Server, Workgroup and Developer Editions, Version 7.30 to 7.31
Informix Dynamic Server, Linux Edition, Version 7.30 to 7.31
INFORMIX-OnLine Dynamic Server, Version 6.0 to 7.24
INFORMIX-OnLine Dynamic Server for Windows NT, Version 7.10 to 7.23
INFORMIX-OnLine Workgroup Server, Version 7.12 and 7.22
INFORMIX-SE, Version 4.1 to 7.22
INFORMIX-OnLine, Version 4.1 to 5.0

December 1999
Part No. 000-6531

Published by Informix® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

© 1999 Informix Corporation. All rights reserved. The following are trademarks of Informix Corporation or its affiliates, one or more of which may be registered in the United States or other jurisdictions:

Answers OnLine™; C-ISAM®; Client SDK™; DataBlade®; Data Director™; Decision Frontier™; Dynamic Scalable Architecture™; Dynamic Server™; Dynamic Server™, Developer Edition™; Dynamic Server™ with Advanced Decision Support Option™; Dynamic Server™ with Extended Parallel Option™; Dynamic Server™ with MetaCube®; Dynamic Server™ with Universal Data Option™; Dynamic Server™ with Web Integration Option™; Dynamic Server™, Workgroup Edition™; Dynamic Virtual Machine™; Extended Parallel Server™; Formation™; Formation Architect™; Formation Flow Engine™; Gold Mine Data Access®; IIF.2000™; i.Reach™; i.Sell™; Illustra®; Informix®; Informix® 4GL; Informix® InquireSM; Informix® Internet Foundation.2000™; InformixLink®; Informix® Red Brick® Decision Server™; Informix Session Proxy™; Informix® Vista™; InfoShelf™; Interforum™; I-Spy™; Mediazation™; MetaCube®; NewEra™; ON-Bar™; OnLine Dynamic Server™; OnLine/Secure Dynamic Server™; OpenCase®; Orca™; PaVER™; Red Brick® and Design; Red Brick® Data Mine™; Red Brick® Mine Builder™; Red Brick® Decisionscape™; Red Brick® Ready™; Red Brick Systems®; Regency Support®; Rely on Red BrickSM; RISQL®; Solution DesignSM; STARindex™; STARjoin™; SuperView®; TARGETindex™; TARGETjoin™; The Data Warehouse Company®; The one with the smartest data wins.™; The world is being digitized. We're indexing it.SM; Universal Data Warehouse Blueprint™; Universal Database Components™; Universal Web Connect™; ViewPoint®; Visionary™; Web Integration Suite™. The Informix logo is registered with the United States Patent and Trademark Office. The DataBlade logo is registered with the United States Patent and Trademark Office.

Documentation Team: Twila Booth, Mary Kraemer, Jennifer Leland, Bonnie Vaughan

GOVERNMENT LICENSE RIGHTS

Software and documentation acquired by or for the US Government are provided with rights as follows:

- (1) if for civilian agency use, with rights as restricted by vendor's standard license, as prescribed in FAR 12.212;
- (2) if for Dept. of Defense use, with rights as restricted by vendor's standard license, unless superseded by a negotiated vendor license, as prescribed in DFARS 227.7202. Any whole or partial reproduction of software or documentation marked with this legend must reproduce this legend.

Table of Contents

Introduction

In This Introduction	3
About This Manual	3
Types of Users	4
Software Dependencies and Database Server Names	5
Assumptions About Your Locale	6
Demonstration Databases	6
New Features That Affect Migration	7
Documentation Conventions	9
Typographical Conventions	10
Icon Conventions	10
Syntax Conventions	13
Command-Line Conventions	17
Sample-Code Conventions	19
Additional Documentation	20
On-Line Manuals	20
Printed Manuals	21
On-Line Help	21
Error Message Documentation	21
Documentation Notes, Release Notes, and Machine Notes	22
Related Reading	23
Compliance with Industry Standards	23
Informix Welcomes Your Comments	24

Section I Overview of Informix Migration

Chapter 1 Database Server Migration

In This Chapter	1-3
Informix Database Server Versions	1-3
Database Server Migration Paths	1-8
Migrating Between Operating Systems	1-11

Chapter 2 Data Migration

In This Chapter	2-3
Migrating a Database or Selected Data	2-3
Migrating Between Different Versions of a Database Server	2-4
Changing Database Servers, Operating Systems, or GLS Locales	2-4
Distributing a Client Application	2-4
Importing Non-Informix Data	2-5
Setting Environment Variables Before Using Utilities	2-5
Choosing the Most Effective Data Migration Tools	2-6
Automatic Data Migration	2-9
The onunload and onload Utilities.	2-9
The dbexport and dbimport Utilities	2-13
LOAD, UNLOAD, and dbload	2-15
The dbschema Utility	2-17
External Tables	2-19
The High-Performance Loader on UNIX.	2-19
The onxfer Utility	2-20
Informix Enterprise Command Center	2-21
Moving Data Between Computers and Dbspaces	2-21
Importing Data from a Non-Informix Source	2-21
Importing Data with Informix Enterprise Gateway Products.	2-22

Chapter 3 Client Software Development Kit (SDK) Compatibility

In This Chapter	3-3
---------------------------	-----

Section II Version 9.x Database Server Migration

Chapter 4 **Migrating Between Dynamic Server 2000 and a 9.14, 7.3, or 7.2 Database Server**

In This Chapter	4-3
Preparing for Migration	4-4
Migration Guidelines.	4-4
Changes That Dynamic Server 2000 Introduced.	4-5
Migration from Universal Server 9.10 to 9.13.	4-7
Migration from OnLine 4.1 or 5.0 or OnLine Dynamic Server 6.0 or 7.1	4-8
Storage-Manager Validation and Installation.	4-8
Upgrading to Dynamic Server 2000 from 9.14, D 7.3x, or 7.2x	4-9
Check and Configure Available Space	4-10
Save Copies of the Current Configuration Files	4-12
Stop Enterprise Replication.	4-13
Close All Transactions and Shut Down the Source Database Server	4-14
Check for Any Open Transactions	4-14
Verify the Integrity of the Data	4-15
Verify the Database Server Mode.	4-16
Make a Final Backup of the Source Database Server	4-16
Verify That the Source Database Server Is Off-Line	4-17
Modify UNIX Kernel Parameters.	4-17
Install Dynamic Server 2000	4-17
Set Environment Variables	4-18
Customize the ONCONFIG Configuration File	4-19
Add Any Communications Support Modules	4-19
For ON-Archive, Update Configuration Files	4-20
Install and Configure Any Datablade Modules	4-20
Initialize Dynamic Server 2000	4-21
Monitor the Upgrade Complete Status	4-21
Update Statistics	4-22
Verify the Integrity of the Data	4-25
Make an Initial Backup of Dynamic Server 2000	4-25
Tune Dynamic Server 2000 for Performance	4-26

Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000	4-26
Determine Whether Reversion Is Possible	4-27
Save Copies of the Configuration Files	4-31
Stop Enterprise Replication	4-31
Verify the Integrity of the Data	4-31
Back Up Dynamic Server 2000	4-31
Remove Version 9.2 Features.	4-32
Run the Reversion Utility	4-32
Modify Configuration Parameters	4-33
Reset Environment Variables.	4-33
Remove Any Communications Support Module Settings	4-33
Reinstall and Start the Target Database Server.	4-34
Update Statistics	4-34
Verify the Integrity of the Data	4-36
Back Up the Target Database Server	4-36
Return the Target Database Server to On-Line Mode	4-36

Chapter 5

Migrating Between Versions of Universal Server

In This Chapter	5-3
Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13	5-3
R-Tree Indexes.	5-4
Tables That Contain Collection Columns	5-4
Reverting to Universal Server 9.10 Through 9.13 from Universal Server 9.14	5-5

Chapter 6

Migrating Between Universal Server and OnLine Dynamic Server 7.2x or 7.1x

In This Chapter	6-3
Preparing for Migration	6-4
Migration Guidelines	6-4
Changes That Universal Server Introduced.	6-5
Reversion to OnLine Dynamic Server 7.2x from Dynamic Server 7.3x	6-8
Upgrading to Universal Server from OnLine Dynamic Server 7.2x or 7.1x	6-8
Check Available Space	6-9
Save Copies of the Current Configuration Files	6-11
Close All Transactions in the Source Database Server	6-12

Put OnLine Dynamic Server in Quiescent Mode	6-12
Verify the Integrity of the Data	6-13
Verify the Mode.	6-13
Back Up OnLine Dynamic Server	6-14
Bring OnLine Dynamic Server Off-Line	6-14
Change UNIX Kernel Parameters	6-15
Install Universal Server	6-15
Set Environment Variables	6-16
Update the ONCONFIG Configuration File	6-16
Add Any Communications Support Modules	6-16
Update the ON-Archive Configuration Files	6-17
Bring Universal Server On-Line	6-17
Verify the Integrity of the Data	6-18
Back Up Universal Server	6-19
Tune Universal Server for Performance	6-19
Install and Configure Any Datablade Modules	6-19
Reverting to OnLine Dynamic Server 7.2x or 7.1x from	
Universal Server	6-20
Determine Whether Reversion Is Possible.	6-21
Save Copies of the Configuration Files	6-21
Verify the Integrity of the Data	6-21
Back Up Universal Server	6-22
Remove New Universal Server Features	6-22
Run the Reversion Utility	6-26
Modify Configuration Parameters	6-26
Reset Environment Variables	6-27
Remove Any Communications Support Module Settings	6-27
Install and Start OnLine Dynamic Server	6-27
Verify the Integrity of the Data	6-27
Back Up OnLine Dynamic Server	6-27
Return OnLine Dynamic Server to On-Line Mode	6-28

Section III Version 8.x Database Server Migration

Chapter 7 Migrating Between Extended Parallel Server and Dynamic Server AD/XP 8.21

In This Chapter	7-3
Preparing for Migration	7-3
Migration Guidelines	7-4
Changes in Extended Parallel Server	7-5
Upgrade and Reversion Paths for 32-Bit and 64-Bit Database Servers.	7-6
Storage-Manager Validation and Installation	7-7
Upgrading to Extended Parallel Server from Dynamic Server AD/XP 8.21.UD4	7-7
Check the Available Space	7-8
Save Copies of the Current Configuration Files	7-9
Take the 8.21.UD4 Database Server to Quiescent Mode.	7-9
Verify the Integrity of the Data	7-10
Make a Final Backup of the 8.21.UD4 Database Server	7-11
Check for Open Transactions and Shut Down the 8.21.UD4 Database Server	7-12
Install Extended Parallel Server.	7-13
Copy the 8.21.UD4 Configuration Files	7-14
Verify Environment Variable Settings on Each Node	7-14
Initialize Extended Parallel Server	7-15
Make an Initial Backup of Extended Parallel Server	7-16
Tune Extended Parallel Server for Performance	7-17
Moving Extended Parallel Server Between 32-Bit and 64-Bit Operating Systems	7-17
Reverting to Dynamic Server AD/XP 8.21.UD4 from Extended Parallel Server	7-17
Save Copies of the Current Configuration Files	7-18
Verify the Integrity of the Data	7-19
Back Up Extended Parallel Server	7-19
Run the Reversion Script in Check Mode	7-19
Remove Features That Extended Parallel Server Introduced	7-20
Leave Extended Parallel Server in On-Line Mode	7-21
Run the Reversion Script	7-22
Reset Environment Variables.	7-25
Install Dynamic Server AD/XP 8.21.UD4	7-25

Copy and Customize Configuration Files	7-25
Initialize Dynamic Server AD/XP 8.21.UD4 Without Initializing Disks	7-26
Verify the Integrity of the Data	7-26
Back Up Dynamic Server AD/XP 8.21.UD4	7-26
Return Dynamic Server AD/XP 8.21.UD4 to On-Line Mode	7-26

Chapter 8 Migrating Between Extended Parallel Server and a 7.3 or 7.2 Database Server

In This Chapter	8-3
Preparing for Migration	8-3
Migration Guidelines	8-4
Changes That Extended Parallel Server Introduces	8-4
Upgrade to OnLine Dynamic Server 7.2x from OnLine Dynamic Server 4.1 to 7.1x	8-6
Storage-Manager Validation and Installation.	8-6
Upgrading to Extended Parallel Server from Dynamic Server 7.3x or OnLine Dynamic Server 7.2x	8-7
Configure and Check Available Space	8-8
Save Copies of the Current Configuration Files	8-9
Close All Transactions in the Source Database Server	8-10
Put the Source Database Server in Quiescent Mode	8-11
Verify the Integrity of the Data	8-11
Verify the Mode.	8-12
Back Up the Source Database Server	8-12
Change UNIX Kernel Parameters	8-13
Edit the Schema File to Suit Extended Parallel Server.	8-13
Set Environment Variables	8-13
Install Extended Parallel Server	8-14
Copy Database Server Utilities to Each Node	8-15
Update the ONCONFIG Configuration File	8-15
Bring Extended Parallel Server On-Line	8-16
Use onutil to Create Cogroups and Dbslices	8-16
Edit the onxfer Configuration File	8-17
Unload the 7.3x or 7.2x Database Schema and Data	8-17
Load the Database Schema and Data	8-17
Make an Initial Backup of Extended Parallel Server	8-17
Tune Extended Parallel Server for Performance.	8-18

Reverting to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x from Extended Parallel Server	8-18
Unloading Data to External Tables.	8-19
Loading Data from External Tables	8-19

Chapter 9 Migrating Between Dynamic Server AD/XP 8.21.UD4 and OnLine XPS 8.11

In This Chapter	9-3
Preparing for Migration	9-3
Migration Guidelines	9-4
Changes in Dynamic Server AD/XP 8.21	9-4
Storage-Manager Validation and Installation	9-5
Upgrading to Dynamic Server AD/XP 8.21.UD4 from OnLine XPS 8.11	9-5
Check the Available Space and System Requirements	9-6
Save Copies of the Current Configuration Files	9-6
Close All Transactions and Shut Down the 8.11 Database Server	9-7
Install Dynamic Server AD/XP 8.21.UD4	9-8
Make an Initial Backup of Dynamic Server AD/XP 8.21.UD4	9-12
Moving Dynamic Server AD/XP 8.21.UD4 Between 32-Bit and 64-Bit Operating Systems	9-12
Reverting to an Earlier 8.x Database Server from Dynamic Server AD/XP 8.21.UD4	9-12

Chapter 10 Migrating Between Dynamic Server AD/XP 8.21 and OnLine Dynamic Server 7.2

In This Chapter	10-3
Preparing for Migration	10-4
Changes That Dynamic Server AD/XP 8.21 Introduced	10-5
Storage-Manager Validation and Installation	10-18
Loading and Unloading Data	10-19
Express Mode	10-19
Deluxe Mode	10-20
Upgrading to Dynamic Server AD/XP 8.21 from OnLine Dynamic Server 7.2x	10-20
Configure and Check Available Space	10-23
Save Copies of the Current Configuration Files	10-24
Close All Transactions in the Source Database Server	10-25
Put OnLine Dynamic Server 7.2x in Quiescent Mode	10-25
Verify the Integrity of the Data	10-26

Verify the Mode.	10-27
Make a Final Backup of OnLine Dynamic Server 7.2x.	10-27
Unload the Data	10-28
Take OnLine Dynamic Server 7.2x Off-Line	10-28
Change UNIX Kernel Parameters	10-29
Install Dynamic Server AD/XP 8.21.	10-29
Set Environment Variables	10-30
Copy Database Server Utilities to Each Node	10-31
Update the ONCONFIG Configuration File	10-31
Update the sqlhosts File or Registry Key	10-31
Update the Backup and Restore Configuration Parameters	10-31
Bring Dynamic Server AD/XP 8.21 On-Line	10-32
Use onutil to Create Cogroups and Dbslices	10-32
Create a Database	10-33
Load Data from External Tables	10-33
Run UPDATE STATISTICS and Build Indexes	10-36
Verify the Integrity of the Data	10-36
Make an Initial Backup of Dynamic Server AD/XP 8.21	10-37
Check Rejection Files	10-37
Recompile ESQL/C Programs.	10-38
Complete Migration	10-38
Reverting to OnLine Dynamic Server 7.2x from Dynamic Server AD/XP 8.21	10-38
Unloading Data to External Tables	10-39
Loading Data from External Tables	10-39

Section IV Version 7.x Database Server Migration

Chapter 11 Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0

In This Chapter	11-5
Changes in Dynamic Server 7.3 and OnLine Dynamic Server 6.0 to 7.2	11-6
Environment Variable Changes in Dynamic Server 7.3	11-6
New Configuration Parameters in Dynamic Server 7.3	11-7
New SQL Reserved Words in Dynamic Server 7.3	11-7
New Features in Dynamic Server 7.3	11-8

Environment Variable Changes in OnLine Dynamic Server 7.2	11-10
Configuration Parameter Changes in OnLine Dynamic Server 7.2	11-12
In-Place ALTER TABLE New in OnLine Dynamic Server 7.24	11-14
Enhancements to sqlhosts File or Registry Key in OnLine Dynamic Server 7.23	11-14
Language Support Changes in OnLine Dynamic Server 6.0 to 7.2	11-15
Environment Variable Changes in OnLine Dynamic Server 7.10.UD1	11-16
Configuration Parameter Changes in OnLine Dynamic Server 7.10.UD1	11-16
Changes to BlobSpace Requirements in OnLine Dynamic Server 7.10.UD1	11-17
Environment Variable Changes in OnLine Dynamic Server 7.1	11-17
Configuration Parameter Changes in OnLine Dynamic Server 7.1	11-18
ON-Archive Changes in OnLine Dynamic Server 7.1	11-18
Environment Variable Changes in OnLine Dynamic Server 6.0	11-19
Preparing for Migration	11-19
Install the Latest Maintenance Release for the Current Version of OnLine Dynamic Server	11-20
Verify and Install a Validated Storage Manager	11-21
Check Available Space and System Requirements	11-21
Save Copies of the Current Configuration Files	11-22
Shut Down the Source Database Server	11-23
Verify the Integrity of the Data	11-24
Back Up the Source Database Server Files	11-25
Begin Migration	11-25
Upgrading to Dynamic Server 7.3	11-26
Save an Output File of SQL Statements for Access Paths	11-27
Bring the Source Database Server Off-Line	11-27
Change UNIX Kernel Parameters	11-27
Install the Target Database Server	11-27
Install Informix Enterprise Command Center	11-28
Customize the Target Database Server Environment	11-29
Update the ONCONFIG Configuration Parameters	11-29
Update the Configuration Files	11-30

Prepare the Target Database Server to Use Enterprise Replication	11-30
Install and Configure SNMP	11-31
Bring the Target Database Server On-Line.	11-32
Update Statistics	11-33
Verify the Integrity of the Data	11-33
Check Space Requirements for BYTE and TEXT Data.	11-33
Make an Initial Backup of the Target Database Server.	11-33
Verify the Access Path of the SQL Statements	11-34
Complete Migration	11-34
Change Database Server Definitions	11-35
Migrating to a GLS Locale	11-35
Reverting to an Earlier Version of OnLine Dynamic Server	11-36
Remove Unsupported SQL Features	11-38
Remove Fragmented Tables (Version 6.0 Only)	11-38
Save System Catalog Information	11-38
Stop Enterprise Replication or Workgroup Replication (Version 7.22 or Later)	11-39
Uninstall the Database Server and Informix Enterprise Command Center	11-40
Modify In-Place ALTER TABLE	11-40
Close All Transactions and Put the Source Database Server in Quiescent Mode	11-41
Verify the Integrity of the Data	11-42
Back Up the Source Database Server	11-42
Run the Reversion Utility	11-43
Remove GLS Features	11-44
Modify Configuration Parameters	11-44
Reset Environment Variables	11-48
Modify the sqlhosts File	11-50
Reinstall the Earlier Version of the Database Server	11-50
Bring the Target Database Server On-Line.	11-50
Rename ON-Archive Files (Version 6.0 Only)	11-51
Restore System Catalog Information (Version 6.0 Only)	11-51
Verify the Integrity of the Data	11-51
Back Up the Target Database Server Files	11-52
Return the Target Database Server to On-Line Mode	11-52
Complete Reversion	11-52

Reverting to OnLine Dynamic Server 7.10.UCx from OnLine	
Dynamic Server 7.10.UDI Through 7.14	11-53
Verify the Integrity of the Data	11-53
Back Up the Target Database Server	11-54
Complete Reversion.	11-54

Chapter 12 **Migrating Between a 7.x or 6.0 Database Server and a 5.x or 4.1 Database Server**

In This Chapter	12-3
Preparing for Migration	12-3
Migration Guidelines	12-4
Changes in OnLine Dynamic Server 6.0	12-4
Operating-System Configuration Issues	12-6
Disk-Utilization Issues	12-12
Database Server Configuration Issues.	12-20
Storage-Manager Validation and Installation	12-23
Upgrading to Dynamic Server 7.3x or OnLine Dynamic Server	
from OnLine 5.0 or 4.1	12-24
Install the Latest Maintenance Release for the Current	
Version of OnLine	12-24
Capture Configuration and Chunk-Layout Information	12-25
Close All Transactions and Make a Final Backup of the	
Source Database Server	12-26
Bring OnLine 5.0 or 4.1 Off-Line	12-28
Install and Configure the Target Database Server.	12-29
Reconfigure the UNIX Operating System	12-31
Bring the Target Database Server On-Line	12-32
Upgrade User-Table Indexes	12-33
Verify the Integrity of the Data	12-37
Make an Initial Backup of the Target Database Server	12-37
Add rowid Columns to Fragmented Tables.	12-37
Complete Migration.	12-38
Convert a Database to a Different Locale	12-38
Reverting to OnLine 5.0 or 4.1	12-39
Save Copies of the Current Configuration Files	12-39
Remove Database Server Users	12-40
Verify the Integrity of the Data	12-40
Back Up the Source Database Server	12-40
Remove Features That Later Versions Introduced	12-40
Run the Reversion Utility	12-41

Prepare the TBCONFIG Configuration File	12-42
Reset Environment Variables	12-43
Bring Up OnLine 5.0 or 4.1 in Quiescent Mode	12-43
Verify the Integrity of the Data	12-43
Back Up OnLine 5.0 or 4.1	12-44
Return Online 5.0 or 4.1 to On-Line Mode.	12-44

Chapter 13 Migrating Between INFORMIX-SE Database Servers and Converting C-ISAM Files

In This Chapter	13-3
Importing and Exporting Data	13-3
Migrating Between Different Versions of SE	13-4
Preparing to Upgrade to SE 7.2 from SE 5.0 or 4.1	13-4
Preparing to Upgrade to SE 7.2 from SE 7.1 or 6.0	13-5
Reverting to an Earlier Version of SE	13-6
Converting C-ISAM Files to SE	13-6

Chapter 14 Migrating Between 7.x Database Server Options and Editions

In This Chapter	14-5
Moving Data Between Database Servers	14-6
Using UNLOAD, dbschema, and LOAD	14-6
Using UNLOAD, dbschema, and dbload	14-8
Migrating to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3	14-8
Save a Copy of the Current Configuration Files	14-9
Verify the Integrity of the Data	14-10
Back Up Dynamic Server 7.3	14-10
Shut Down Dynamic Server 7.3	14-10
Install and Configure Dynamic Server, Workgroup and Developer Editions	14-12
Verify Port Numbers and Services File	14-12
Customize the Environment for Dynamic Server, Workgroup and Developer Editions	14-12
Bring Dynamic Server, Workgroup and Developer Editions, On-Line	14-13
Verify the Integrity of the Data	14-13
Back Up Dynamic Server, Workgroup and Developer Editions	14-13
Update Statistics	14-14

Complete Migration.	14-14
Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions	14-14
Migrating to Dynamic Server 7.3 from Dynamic Server, W/D . . .	14-15
Save a Copy of the Current Configuration Files	14-16
Verify the Integrity of the Data	14-16
Back Up Dynamic Server, Workgroup and Developer Editions.	14-17
Shut Down Dynamic Server, Workgroup and Developer Editions.	14-17
Install and Configure Dynamic Server 7.3	14-18
Verify Port Numbers and Services File	14-19
Customize the Environment for Dynamic Server 7.3	14-19
Bring Dynamic Server 7.3 On-Line.	14-19
Verify the Integrity of the Data	14-20
Back Up Dynamic Server 7.3.	14-20
Update Statistics	14-20
Complete Migration.	14-20
Adapt Your Programs for Dynamic Server 7.3.	14-21
Migrating Between Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server	14-21
Using Different Server and Client Configurations	14-22
Preparing for Migration	14-23
Upgrading to Dynamic Server, Workgroup and Developer Editions, Version 7.3	14-27
Changing Database Server Definitions on Windows NT	14-37
Migrating to a GLS Locale	14-38
Reverting the Database Server and IECC	14-38
Moving Data Between SE and Dynamic Server 7.3 or Its Editions . .	14-45
Moving Data to Dynamic Server 7.3 or Its Editions from SE . .	14-45
Moving Data to SE from Dynamic Server or Its Editions	14-50

Section V Migration Between Operating Systems

Chapter 15 Moving a Database Server to a Different Operating System

In This Chapter	15-3
Choosing a Migration Method	15-3
Adjusting Database Tables for File-System Variations	15-5
Moving Data to a Database Server on a Different Operating System	15-5
Using the Migration Utilities	15-6
Adapting Your Programs for UNIX or Windows NT	15-7
Completing Migration	15-7

Chapter 16 Migrating Between Dynamic Server 7.3 and Its Editions in Different Environments

In This Chapter	16-3
Choosing a Migration Method	16-4
Migrating Data to Dynamic Server, W/D from 7.3 on a Different Operating System	16-5
Save Copies of the Current Configuration Files	16-6
Verify the Integrity of the Data	16-6
Back Up Dynamic Server 7.3	16-7
Export Data from Dynamic Server 7.3	16-7
Shut Down Dynamic Server 7.3	16-11
Install and Configure Dynamic Server, Workgroup and Developer Editions	16-12
Verify Port Numbers and the Services File	16-12
Customize the Database Server Environment	16-12
Bring Dynamic Server, Workgroup and Developer Editions, On-Line	16-13
Import Data into Dynamic Server, Workgroup and Developer Editions	16-13
Verify the Integrity of the Data	16-14
Make an Initial Backup of Dynamic Server, Workgroup and Developer Editions	16-14
Run UPDATE STATISTICS	16-14
Complete Migration	16-15
Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions	16-15

Migrating Data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on a Different Operating System	16-15
Save Copies of the Current Configuration Files	16-16
Verify the Integrity of the Data	16-17
Back Up Dynamic Server, Workgroup and Developer Editions	16-17
Export Data from Dynamic Server, Workgroup and Developer Editions	16-17
Shut Down Dynamic Server, Workgroup and Developer Editions	16-20
Install and Configure Dynamic Server 7.3	16-21
Verify Port Numbers and the Services File	16-21
Customize the Database Server Environment	16-21
Bring Dynamic Server 7.3 On-Line.	16-22
Import the Data into the Target Database Server	16-22
Verify the Integrity of the Data	16-22
Back Up the Target Database Server	16-23
Run UPDATE STATISTICS	16-23
Complete Migration.	16-23
Adapt Your Programs for Dynamic Server 7.3.	16-24

Section VI Locale Changes

Chapter 17 Understanding Language Support in Informix Database Servers

In This Chapter	17-3
Informix Language Support	17-4
Global Language Support	17-4
Native Language Support	17-9
Asian Language Support	17-14
Language Support in Database Servers	17-20
Dynamic Server 2000 and Universal Server.	17-21
Extended Parallel Server, Dynamic Server AD/XP, and OnLine XPS.	17-22
Dynamic Server 7.3 and OnLine Dynamic Server.	17-22
INFORMIX-SE	17-23
Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server.	17-24

Chapter 18	Converting to GLS	
	In This Chapter	18-3
	Determining the Existing Locale Support	18-3
	Finding the Current Database Locale	18-4
	Finding Available GLS Locales on UNIX	18-4
	Finding Available GLS Locales on Windows NT	18-5
	Verifying GLS Database Locales	18-5
	Upgrading to GLS from NLS	18-6
	Determining the Locale of the NLS Database	18-6
	Determining Whether a Compatible GLS Locale Exists	18-8
	Opening an NLS Database	18-11
	Upgrading to GLS from ALS	18-13
	Migrating from Version 6.x ALS Products	18-13
	Migrating from Version 5.x ALS Products	18-15
	Migrating from Version 4.x ALS Products	18-17
	Migrating from Version 4.x ASCII Japanese Products	18-19
 Chapter 19	 Reverting from GLS	
	In This Chapter	19-3
	Using the onmode Utility for Reversion	19-3
	Reverting to NLS from GLS	19-3
	Reverting to an NLS Database Server	19-4
	Reverting to an NLS Version of the SE Database Server	19-5
	Reverting to ALS from GLS	19-5
	Reverting to Version 6.x ALS	19-6
	Reverting to Version 5.x ALS	19-7
	Reverting to Version 4.x ALS or ASCII Products	19-8

Section VII Data Migration Utilities

Chapter 20 Using the dbexport and dbimport Utilities

In This Chapter	20-3
Syntax of the dbexport Command	20-5
Syntax of the dbimport Command	20-11
Simple Large Objects	20-19
Database Locale Changes	20-20

Chapter 21 Using the dbload Utility

In This Chapter	21-3
Syntax of the dbload Command	21-4
Command File for dbload	21-9
Command File to Load Complex Data Types	21-21

Chapter 22 Using the dbschema Utility

In This Chapter	22-3
Syntax of the dbschema Command	22-4
Database Schema Creation	22-6
Server-Specific Information	22-8
User-Defined and Complex Data Types	22-9
Synonym Creation	22-11
Privileges	22-12
Table, View, or Procedure Creation	22-14
Table Information	22-16
Role Creation	22-17
Distribution Information for Tables	22-18
DB-Access Input from dbschema Output	22-22

Chapter 23 Using the LOAD and UNLOAD Statements

In This Chapter	23-3
Syntax of the UNLOAD Statement	23-4
Syntax of the LOAD Statement	23-5

Chapter 24	Using the onmode Utility	
	In This Chapter	24-3
	Use of the onmode -b Command for Reversion	24-4
	Preparation for Reversion	24-4
	Syntax of the onmode -b Command	24-5
Chapter 25	Using the onunload and onload Utilities	
	In This Chapter	25-3
	How onunload and onload Work	25-4
	Syntax of the onunload Command	25-5
	Syntax of the onload Command	25-10
	Constraints That Affect onload and onunload	25-13
	Steps for Using onunload and onload	25-16
Chapter 26	Using the onxfer Utility	
	In This Chapter	26-3
	Environment Variables for onxfer	26-4
	Syntax of the onxfer Command	26-5
	Configuration File for onxfer	26-7
	Guidelines and Restrictions for Using onxfer	26-17
Appendix A	Database Server Environment Variables	
Appendix B	Upgrade and Reversion Messages	
	Index	

Introduction

In This Introduction	3
About This Manual.	3
Types of Users	4
Software Dependencies and Database Server Names	5
Assumptions About Your Locale.	6
Demonstration Databases	6
New Features That Affect Migration	7
Documentation Conventions	9
Typographical Conventions	10
Icon Conventions	10
Comment Icons	11
Feature, Product, and Platform Icons	11
Syntax Conventions	13
Syntax Elements in a Statement Path	14
How to Read a Syntax Diagram.	16
Command-Line Conventions	17
Syntax Elements in a Command-Line Path	17
How to Read a Command-Line Diagram	19
Sample-Code Conventions	19
Additional Documentation	20
On-Line Manuals	20
Printed Manuals	21
On-Line Help	21
Error Message Documentation	21
Documentation Notes, Release Notes, and Machine Notes	22
Related Reading	23

Compliance with Industry Standards.	23
Informix Welcomes Your Comments	24

In This Introduction

This introduction provides an overview of the information in this manual and describes the conventions it uses.

About This Manual

This manual describes the migration procedures that you can use to upgrade to or revert from one Informix database server to another, move data from one database server to another, change locales, or move a database server to a different operating system. This manual explains migration among Version 9.x, Version 8.x, Version 7.x, and earlier Informix database servers. It also explains how to use the **dbschema**, **dbload**, **dbunload**, **dbexport**, **dbimport**, **onunload**, and **onxfer** data migration utilities and the LOAD and UNLOAD SQL statements.

Migration includes upgrading to a newer version or different kind of Informix database server, reversion to a previous database server, movement of a database server to a different operating system, and movement of data between database servers. Upgrading or reversion often involves changing connectivity information in the **sqlhosts** file or registry key, host environment variables, configuration parameters, and other database-server-specific features.

Types of Users

This manual is for the following users:

- Database users
- Database administrators
- Database server administrators
- System administrators
- Database-application programmers
- Backup operators
- Performance engineers

Before reading this manual, you probably need the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming
- Some experience with database server administration, operating system administration, or network administration

If you have limited experience with relational databases, SQL, or your operating system, refer to the *Getting Started* manual for your database server for a list of supplementary titles.

Software Dependencies and Database Server Names

This guide covers migration from one Informix database server to another or from one version of an Informix database server to another. This guide uses the following database server names, including abbreviations, short names, and complete database server names with version numbers.

Abbreviation	Short Database Server Name	Complete Database Server Name
IDS 2000	Dynamic Server 2000	Informix Dynamic Server 2000, Version 9.2
IDS 7.3	Dynamic Server 7.3	Informix Dynamic Server, Version 7.30 or Version 7.31
XPS 8.3	Extended Parallel Server	Informix Extended Parallel Server, Version 8.30
IDS W/D	Dynamic Server, Workgroup and Developer Editions	Informix Dynamic Server, Workgroup and Developer Editions, Version 7.30 or Version 7.31
LE	Dynamic Server, Linux Edition	Informix Dynamic Server, Linux Edition, Version 7.30 or Version 7.31
IUS	Universal Server	INFORMIX-Universal Server, Version 9.14, Version 9.13, Version 9.12, Version 9.11, or Version 9.10
IDS AD/XP	Dynamic Server with AD and XP Options	Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options, Version 8.21
XPS 8.11	OnLine XPS	INFORMIX-OnLine XPS, Version 8.10 or Version 8.11
ODS	OnLine Dynamic Server	INFORMIX-OnLine Dynamic Server, Version 7.24, Version 7.23, Version 7.22, Version 7.21, Version 7.20, Version 7.14, Version 7.13, Version 7.12, Version 7.11, Version 7.10, or Version 6.0 INFORMIX-OnLine Dynamic Server for Windows NT, Version 7.23, Version 7.22, Version 7.20, Version 7.12, or Version 7.10
OWS	OnLine Workgroup Server	INFORMIX-OnLine Workgroup Server, Version 7.22 or Version 7.12
SE	SE	INFORMIX-SE, Version 7.22, Version 7.2x, Version 7.1x, Version 6.0, Version 5.0, or Version 4.1
OL	OnLine	INFORMIX-OnLine, Version 5.0 or Version 4.1

In this guide, a short database server name followed by a version number refers only to that version, or set of versions, of the database server. For example, Universal Server 9.14 refers only to INFORMIX-Universal Server, Version 9.14, and OnLine Dynamic Server 7.2 refers to any of the following database servers: INFORMIX-OnLine Dynamic Server, Version 7.24, Version 7.23, Version 7.22, Version 7.21, or Version 7.20 or INFORMIX-OnLine Dynamic Server for Windows NT, Version 7.23, Version 7.22, or Version 7.20.

Assumptions About Your Locale

Informix products can support many languages, cultures, and code sets. A single environment, called a Global Language Support (GLS) locale, brings all culture-specific information together.

The examples in this manual use the default GLS locale, **en_us.8859-1**. This locale supports U.S. English format conventions for dates, times, and currency. In addition, this locale supports the ISO 8859-1 code set, which includes the ASCII code set plus many 8-bit characters such as é, è, and ñ.

If you plan to use nondefault characters in your data or your SQL identifiers, or if you want to conform to the nondefault collation rules of character data, you need to specify the appropriate nondefault locale.

For instructions on how to specify a nondefault locale, additional syntax, and other considerations related to GLS locales, see the *Informix Guide to GLS Functionality*.

Demonstration Databases

The DB-Access utility, which Informix provides with its database server products, includes one or more of the following demonstration databases:

- The **stores_demo** database illustrates a relational schema with information about a fictitious wholesale sporting-goods distributor. Many examples in Informix manuals are based on the **stores_demo** database.
- The **sales_demo** database illustrates a dimensional schema for data-warehousing applications. For conceptual information about dimensional data modeling, see the *Informix Guide to Database Design and Implementation*. ♦

- The **superstores_demo** database illustrates an object-relational schema. The **superstores_demo** database contains examples of extended data types, type and table inheritance, and user-defined routines. ♦

For information about how to create and populate the demonstration databases, see the *DB-Access User's Manual*. For descriptions of the databases and their contents, see the *Informix Guide to SQL: Reference*.

The scripts that you use to install the demonstration databases reside in the **\$INFORMIXDIR/bin** directory on UNIX platforms and in the **%INFORMIXDIR%\bin** directory in Microsoft Windows environments.

New Features That Affect Migration

For a comprehensive list of new database server features, see your release notes. This section lists new features relevant to this manual.

Dynamic Server 2000 has all the Universal Server 9.14 features plus the following new features that affect migration:

- Support for parallel user-defined routine (UDR) execution in the High-Performance Loader (HPL)
- Long identifiers
 - 128-character identifiers
 - 32-character user names
- Environment variables
 - **IFX_LONGID**
 - **STMT_CACHE**
 - **STMT_CACHE_DEBUG**
- New SQL reserved words

- Features from Dynamic Server 7.3
 - ❑ ALTER FRAGMENT ATTACH/DETACH enhancements
 - ❑ In-place ALTER TABLE MODIFY/DROP (for built-in types)
 - ❑ Restartable restore
 - ❑ External backup and restore
 - ❑ Application migration features:
 - UPPER, LOWER, and INITCAP functions for case-insensitive search (for built-in types)
 - REPLACE, SUBSTR, LPAD, and RPAD functions for string manipulation (for built-in types)
 - UNION operator in CREATE VIEW statement
 - CASE expression
 - NVL and DECODE functions
 - TO_CHAR and TO_DATE date-conversion functions (for built-in types)
 - IFX_UPDDESC environment variable to describe an UPDATE statement
 - EXECUTE PROCEDURE syntax to update triggering columns
 - New arguments to the **dbinfo()** function to obtain the hostname and version of the database server
 - ❑ Manageability features:
 - Informix Storage Manager (ISM) to manage the storage devices and media that contain backups
 - Additional information for onsnmp Management Information Bases
 - ❑ New SQL reserved words

Extended Parallel Server has the following new features that affect migration:

- Data-migration enhancements
 - **onxfer** data-movement utility to Extended Parallel Server from Dynamic Server, Version 7.31 or Version 7.30, or OnLine Dynamic Server
 - Reversion utility for migrating to Dynamic Server with AD and XP Options, Version 8.21, from Extended Parallel Server
- Configuration enhancements
 - Configurable page size
 - Large chunk size (greater than 2 gigabytes)
 - 64-bit very large memory (VLM)
 - **NODEFAC**, **DBCENTURY**, **SOURCE_REMOTE_SHELL**, and **XFER_CONFIG** environment variables

Documentation Conventions

This section describes the conventions that this manual uses. These conventions make it easier to gather information from this and other volumes in the documentation set.

The following conventions are discussed:

- Typographical conventions
- Icon conventions
- Syntax conventions
- Command-line conventions
- Sample-code conventions

Typographical Conventions

This manual uses the following conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All primary elements in a programming language statement (keywords) appear in uppercase letters in a serif font.
<i>italics</i> italics <i>italics</i>	Within text, new terms and emphasized words appear in italics. Within syntax and code examples, variable values that you are to specify appear in italics.
boldface boldface	Names of program entities (such as classes, events, and tables), environment variables, file and pathnames, and interface elements (such as icons, menu items, and buttons) appear in boldface.
<code>monospace</code> <code>monospace</code>	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
◆	This symbol indicates the end of one or more product- or platform-specific paragraphs.
→	This symbol indicates a menu item. For example, “Choose Tools→Options ” means choose the Options item from the Tools menu.






***Tip:** When you are instructed to “enter” characters or to “execute” a command, immediately press RETURN after the entry. When you are instructed to “type” the text or to “press” other keys, no RETURN is required.*

Icon Conventions

Throughout this guide, several different icons identify comment text or text that applies only to a specific product, feature, or computer platform. This section describes these icons.






Comment Icons

Comment icons identify three types of information, as the following table describes. This information always appears in *italics*.

Icon	Label	Description
	<i>Warning:</i>	Identifies paragraphs that contain vital instructions, cautions, or critical information
	<i>Important:</i>	Identifies paragraphs that contain significant information about the feature or operation that is being described
	<i>Tip:</i>	Identifies paragraphs that offer additional details or shortcuts for the functionality that is being described

Feature, Product, and Platform Icons

Feature, product, and platform icons identify paragraphs that contain feature-specific, product-specific, or platform-specific information.

Icon	Description
	Identifies information or syntax that is specific to Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options
	Identifies information that is specific to an Asian Language Support (ALS) database or application
	Identifies information that is valid only for DB-Access
	Identifies information that is specific to the Informix ESQL/C product
	Identifies information that relates to the Informix Global Language Support (GLS) feature.

(1 of 2)

Icon	Description
IDS	Identifies information that is specific to Informix Dynamic Server 2000 or Informix Dynamic Server 7.3
IDS 2000	Identifies information that is specific to Informix Dynamic Server 2000
IUS	Identifies information that is specific to INFORMIX-Universal Server
LE	Identifies information that is specific to Informix Dynamic Server, Linux Edition
NLS	Identifies information that is specific to a Native Language Support (NLS) database or application
ODS	Identifies information that is specific to INFORMIX-OnLine Dynamic Server
OWS	Identifies information that is specific to INFORMIX-OnLine Workgroup Server
SE	Identifies information that is specific to INFORMIX-SE
UNIX	Identifies information that is specific to UNIX platforms
W/D	Identifies information that is specific to Informix Dynamic Server, Workgroup and Developer Editions
WIN NT	Identifies information that is specific to the Windows NT environment
WIN NT/95	Identifies information that is specific to Windows NT and Windows 95 environments
XPS 8.3	Identifies information that is specific to Extended Parallel Server
XPS 8.11	Identifies information that is specific to INFORMIX-OnLine XPS

(2 of 2)

These icons can apply to an entire section or to one or more paragraphs within a section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of feature-, product-, or platform-specific information that appears within one or more paragraphs within a section.

Syntax Conventions

This section describes conventions for syntax diagrams. Each diagram displays the sequences of required and optional keywords, terms, and symbols that are valid in a given statement or segment, as [Figure 1](#) shows.

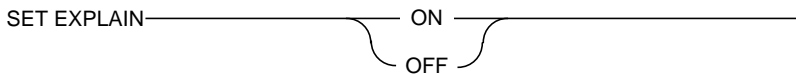


Figure 1
Example of a
Simple
Syntax Diagram







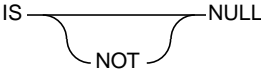
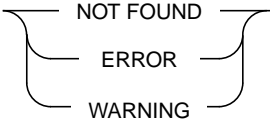
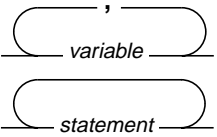
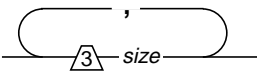

Each syntax diagram begins at the upper-left corner and ends at the upper-right corner with a vertical terminator. Between these points, any path that does not stop or reverse direction describes a possible form of the statement.

Syntax elements in a path represent terms, keywords, symbols, and segments that can appear in your statement. The path always approaches elements from the left and continues to the right, except in the case of separators in loops. For separators in loops, the path approaches counterclockwise. Unless otherwise noted, at least one blank character separates syntax elements.

Syntax Elements in a Statement Path

You might encounter one or more of the following syntax elements in the diagram of a statement path.

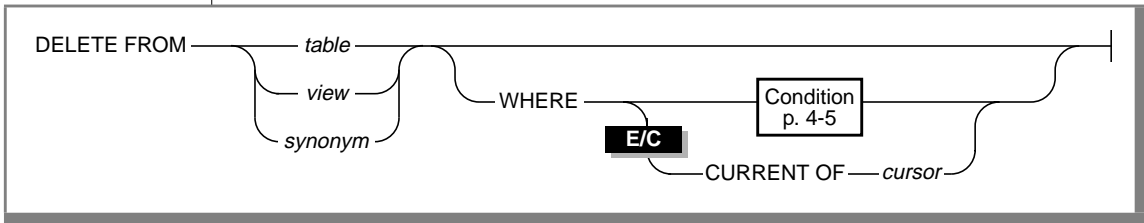
Element	Description
KEYWORD	A word in UPPERCASE letters is a keyword. You must spell the word exactly as shown; however, you can use either uppercase or lowercase letters.
(, ; @ + * - /)	Punctuation and other nonalphanumeric characters are literal symbols that you must enter exactly as shown.
' '	Single quotes are literal symbols that you must enter as shown.
<i>variable</i>	A word in <i>italics</i> represents a value that you must supply. A table immediately following the diagram explains the value.
<div>ADD Clause p. 3-288</div> <div>ADD Clause</div>	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
<div>Back to ADD Clause p. 1-14</div>	A reference in a box in the upper-right corner of a subdiagram refers to the next higher-level diagram of which this subdiagram is a member.
<div>E/C</div>	<p>An icon is a warning that this path is valid only for some products, or only under certain conditions. Characters on the icons indicate what products or conditions support the path.</p> <p>These icons might appear in a syntax diagram:</p> <div><div>XPS 8.3</div><div>This path is valid only for Extended Parallel Server.</div></div> <div><div>AD/XP</div><div>This path is valid only for Dynamic Server with AD and XP Options.</div></div>

Element	Description
	This path is valid only for DB-Access.
	This path is valid only for ESQL/C.
	This path is valid only for Dynamic Server 2000 or Dynamic Server 7.3.
	A shaded option is the default action.
	Syntax within a pair of arrows is a subdiagram.
	The vertical line terminates the syntax diagram.
	A branch below the main path indicates an optional path. (Any term on the main path is required, unless a branch can circumvent it.)
	A set of multiple branches indicates that a choice among more than two different paths is available.
	A loop indicates a path that you can repeat. Punctuation along the top of the loop indicates the separator symbol for list items. If no symbol appears, a blank space is the separator.
	A gate () on a path indicates that you can only use that path the indicated number of times, even if it is part of a larger loop. You can specify <i>size</i> no more than three times within this statement segment.

How to Read a Syntax Diagram

Figure 1 shows a syntax diagram that uses most of the path elements that the previous table lists.

Figure 1
Example of a Syntax Diagram



To use this diagram to construct a statement, start at the top left with the keyword `DELETE FROM`. Then follow the diagram to the right, proceeding through the options that you want.

Figure 1 illustrates the following steps:

1. Type `DELETE FROM`.
2. You can delete a table, view, or synonym:
 - Type the table name, view name, or synonym, as you desire.
 - You can type `WHERE` to limit the rows to delete.
 - If you type `WHERE` and you are using DB-Access or the SQL Editor, you must include the Condition clause to specify a condition to delete. To find the syntax for specifying a condition, go to the “Condition” segment on the specified page.
 - If you are using ESQL/C, you can include either the Condition clause to delete a specific condition or the `CURRENT OF cursor` clause to delete a row from the table.
3. Follow the diagram to the terminator.
Your `DELETE` statement is complete.

Command-Line Conventions

This section defines and illustrates the format of commands that are available in Informix products. These commands have their own conventions, which might include alternative forms of a command, required and optional parts of the command, and so forth.

Each diagram displays the sequences of required and optional elements that are valid in a command. A diagram begins at the upper-left corner with a command. It ends at the upper-right corner with a vertical line. Between these points, you can trace any path that does not stop or back up. Each path describes a valid form of the command. You must supply a value for words that are in italics.

Syntax Elements in a Command-Line Path

You might encounter one or more of the following syntax elements in the diagram of a command-line path.

Element	Description
command	This required element is usually the product name or other short word that invokes the product or calls the compiler or preprocessor script for a compiled Informix product. It might appear alone or precede one or more options. You must spell a command exactly as shown and use lowercase letters.
<i>variable</i>	A word in italics represents a value that you must supply, such as a database, file, or program name. A table following the diagram explains the value.
-flag	A flag is usually an abbreviation for a function, menu, or option name, or for a compiler or preprocessor argument. You must enter a flag exactly as shown, including the preceding hyphen.
.ext	A filename extension, such as .sql or .cob , might follow a variable that represents a filename. Type this extension exactly as shown, immediately after the name of the file. The extension might be optional in certain products.

(1 of 2)

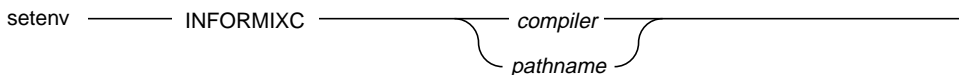
Element	Description
(. , ; + * - /)	Punctuation and mathematical notations are literal symbols that you must enter exactly as shown.
' '	Single quotes are literal symbols that you must enter as shown.
<div>Privileges p. 5-17</div> <div>Privileges</div>	A reference in a box represents a subdiagram. Imagine that the subdiagram is spliced into the main diagram at this point. When a page number is not specified, the subdiagram appears on the same page.
— ALL —	A shaded option is the default action.
→ →	Syntax within a pair of arrows indicates a subdiagram.
—	The vertical line terminates the command.
-f — OFF — ON —	A branch below the main path indicates an optional path. (Any term on the main path is required, unless a branch can circumvent it.)
— , — variable	A loop indicates a path that you can repeat. Punctuation along the top of the loop indicates the separator symbol for list items.
— , — 3 size	A gate ($\sqrt{3}$) on a path indicates that you can only use that path the indicated number of times, even if it is part of a larger loop. You can specify <i>size</i> no more than three times within this statement segment.

(2 of 2)

How to Read a Command-Line Diagram

Figure 2 shows a command-line diagram that uses some of the elements that are listed in the previous table.

Figure 2
Example of a Command-Line Diagram



To construct a command correctly, start at the top left with the command. Follow the diagram to the right, including the elements that you want. The elements in the diagram are case sensitive.

Figure 2 illustrates the following steps:

1. Type `setenv`.
2. Type `INFORMIXC`.
3. Supply either a compiler name or a pathname.
After you choose *compiler* or *pathname*, you come to the terminator.
Your command is complete.
4. Press RETURN to execute the command.

Sample-Code Conventions

Examples of SQL code occur throughout this manual. Except where noted, the code is not specific to any single Informix application development tool. If only SQL statements are listed in the example, they are not delimited by semicolons. For instance, you might see the code in the following example:

```

CONNECT TO stores_demo
...

DELETE FROM customer
      WHERE customer_num = 121
...

COMMIT WORK
DISCONNECT CURRENT
  
```



To use this SQL code for a specific product, you must apply the syntax rules for that product. For example, if you are using DB-Access, you must delimit multiple statements with semicolons. If you are using an SQL API, you must use EXEC SQL at the start of each statement and a semicolon (or other appropriate delimiter) at the end of the statement.

Tip: *Ellipsis points in a code example indicate that more code would be added in a full application, but it is not necessary to show it to describe the concept being discussed.*

For detailed directions on using SQL statements for a particular application development tool or SQL API, see the manual for your product.

Additional Documentation

For additional information, you might want to refer to the following types of documentation:

- On-line manuals
- Printed manuals
- On-line help
- Error message documentation
- Documentation notes, release notes, and machine notes
- Related reading

On-Line Manuals

An Answers OnLine CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print on-line manuals, see the installation insert that accompanies Answers OnLine.

Informix on-line manuals are also available on the following Web site:

www.informix.com/answers

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order:

- The documentation that you need
- The quantity that you need
- Your name, address, and telephone number

WIN NT

On-Line Help

Informix provides on-line help with each graphical user interface (GUI) that displays information about those interfaces and the functions that they perform. Use the help facilities that each GUI provides to display the on-line help.

Error Message Documentation

Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions.

To read error messages and corrective actions on UNIX, use one of the following utilities.

Utility	Description
finderr	Displays error messages on line
rofferr	Formats error messages for printing

◆

UNIX

WIN NT/95

To read error messages and corrective actions in Microsoft Windows environments, use the **Informix Find Error** utility. To display this utility, choose **Start→Programs→Informix** from the Task Bar. ♦

Instructions for using the preceding utilities are available in Answers OnLine. Answers OnLine also provides a listing of error messages and corrective actions in HTML format.

Documentation Notes, Release Notes, and Machine Notes

In addition to printed documentation, the following sections describe the on-line files that supplement the information in this manual. Please examine these files before you begin using your database server. They contain vital information about application and performance issues.

UNIX

On UNIX, the following on-line files appear in the **\$INFORMIXDIR/release/en_us/0333** directory. Replace **x.y** in the filenames with the version number of your database server.

On-Line File	Purpose
MIGRATEDOC_x.y	The documentation notes file for your version of this manual describes topics that are not covered in the manual or that were modified since publication.
SERVERS_x.y	The release notes file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.
IDS_x.y or XPS_x.y	The machine notes file describes any special actions that you must take to configure and use Informix products on your computer. Machine notes are named for the product described.

♦

WIN NT

The following items appear in the **Informix** folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar.

Program Group Item	Description
Documentation Notes	This item includes additions or corrections to manuals and information about features that might not be covered in the manuals or that have been modified since publication.
Release Notes	This item describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds.

The machine notes do not apply to Windows NT. ♦

Related Reading

The following publications provide additional information about the topics that this manual discusses:

- *Your Administrator's Guide*
- *The Administrator's Reference*
- *Your Installation Guide*

For a list of publications that provide an introduction to database servers and operating system environments, refer to your *Getting Started* manual.

Compliance with Industry Standards

The American National Standards Institute (ANSI) has established a set of industry standards for SQL. Informix SQL-based products are fully compliant with SQL-92 Entry Level (published as ANSI X3.135-1992), which is identical to ISO 9075:1992. In addition, many features of Informix database servers comply with the SQL-92 Intermediate and Full Level and X/Open SQL CAE (common applications environment) standards.

Informix Welcomes Your Comments

Let us know what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any corrections or clarifications that you would find useful. Please include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Send electronic mail to us at the following address:

`doc@informix.com`

The **doc** alias is reserved exclusively for reporting errors and omissions in our documentation.

We appreciate your suggestions.

Overview of Informix Migration

- Chapter 1 Database Server Migration
- Chapter 2 Data Migration
- Chapter 3 Client Software Development Kit (SDK) Compatibility

Section I



Database Server Migration

In This Chapter	1-3
Informix Database Server Versions	1-3
Database Server Migration Paths	1-8
Migrating Between Operating Systems	1-11

In This Chapter

This chapter provides an overview of database server migration, including the following topics:

- Informix database server versions
- Database server migration paths
- Migration between operating systems

Informix Database Server Versions

To become more familiar with the Informix client-server environment, read *Getting Started with Informix Database Server Products*. It discusses the differences between Informix database servers and network and server configurations. Also, read the *Getting Started* manual for your database server.

[Figure 1-1 on page 1-4](#) lists the Informix database servers by name and version number, with the operating systems in which you can use them. [Figure 1-1](#) also gives the short form of the database server name in parenthesis.

Figure 1-1
Informix Database Servers

Database Server Name	Version Number(s)	Operating System(s)
Informix Dynamic Server 2000 (Dynamic Server 2000)	9.20	UNIX, Windows NT
INFORMIX-Universal Server (Universal Server)	9.14 9.12	UNIX, Windows NT
INFORMIX-Universal Server (Universal Server)	9.13 9.11 9.10	UNIX
Informix Extended Parallel Server (Extended Parallel Server)	8.30	UNIX
Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options (Dynamic Server with AD and XP Options)	8.21	UNIX, Windows NT
INFORMIX-OnLine XPS (OnLine XPS)	8.11 8.10	UNIX
Informix Dynamic Server 7.3 (Dynamic Server 7.3) or Informix Dynamic Server, Workgroup and Developer Editions (Dynamic Server, Workgroup and Developer Editions)	7.30 7.31	UNIX, Windows NT
Informix Dynamic Server, Linux Edition (Dynamic Server, Linux Edition)	7.30 7.31	Linux
INFORMIX-OnLine Dynamic Server (OnLine Dynamic Server)	7.24 7.23 7.22 7.21 7.20 7.14 7.13 7.12 7.11 7.10.UD1 7.10 6.0	UNIX

(1 of 2)

Database Server Name	Version Number(s)	Operating System(s)
INFORMIX-OnLine Dynamic Server for Windows NT (OnLine Dynamic Server)	7.24	Windows NT
	7.23	
	7.22	
	7.20	
	7.12	
INFORMIX-OnLine Workgroup Server (OnLine Workgroup Server)	7.10	Windows NT
	7.22	
	7.12	
	7.20	
	7.24	
INFORMIX-OnLine Workgroup Server (OnLine Workgroup Server)	7.12	UNIX, Windows NT
INFORMIX-SE (SE)	7.22	UNIX, Windows NT
	6.0	
	5.0	
	4.1	
INFORMIX-OnLine (OnLine)	5.0	UNIX
	4.1	

(2 of 2)

For installation instructions, refer to your *Installation Guide*. For information on how to administer and configure your database server, refer to your *Administrator's Guide*.

UNIX

Figure 1-2 lists the source and target versions for migration of database servers that Informix supports on UNIX. You might have previously purchased another version of an Informix database server. If you have a database server version that Figure 1-2 does not list, you can migrate to a version that Informix supports.

To migrate between UNIX and Windows NT, see Chapter 15, “Moving a Database Server to a Different Operating System.”

Figure 1-2
Migrating Between Database Servers on UNIX

Source Version	Target Version																			
	9.20.U	9.14.U	9.13.U	9.12.U	9.11.U	9.10.U	8.30.U	8.21.UD4	8.21.U	8.11.U	8.10.U	7.31.U	7.30.U	7.24.U	7.23.U	7.22.U	7.21.U	7.20.U	7.14.U	7.13.U
9.20.U	•	•										•	•	•	•	•	•	•		
9.14.U	•	•	•	•	•	•								•	•	•	•	•	•	•
9.13.U		•	•	•	•	•								•	•	•	•	•	•	•
9.12.U		•	•	•	•	•								•	•	•	•	•	•	•
9.11.U		•	•	•	•	•								•	•	•	•	•	•	•
9.10.U		•	•	•	•	•								•	•	•	•	•	•	•
8.30.U							•	•				•	•	•	•	•	•	•		
8.21.UD4							•	•	•	•										
8.21.U								•	•											
8.11.U								•		•	•									
8.10.U										•	•									
7.31.U	•						•			•	•	•	•	•	•	•	•	•	•	•
7.30.U	•						•			•	•	•	•	•	•	•	•	•	•	•
7.24.U	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•
7.23.U	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•
7.22.U	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•
7.21.U	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•
7.20.U	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•
7.14.U		•	•	•	•	•						•	•	•	•	•	•	•	•	•
7.13.U		•	•	•	•	•						•	•	•	•	•	•	•	•	•
7.12.U		•	•	•	•	•						•	•	•	•	•	•	•	•	•
7.11.U		•	•	•	•	•						•	•	•	•	•	•	•	•	•
7.10.U		•	•	•	•	•						•	•	•	•	•	•	•	•	•
6.0x.U												•	•	•	•	•	•	•	•	•
5.0x.U												•	•	•	•	•	•	•	•	•
4.1x.U												•	•	•	•	•	•	•	•	•



WIN NT

Figure 1-3 lists the source and target versions for migration of database servers that Informix supports on Windows NT.

Figure 1-3
Migrating Between Database Servers on Windows NT

Source Version	Target Version											
	9.20.T	9.14.T	9.12.T	8.21.T	7.31.T	7.30.T	7.24.T	7.23.T	7.22.T	7.20.T	7.12.T	7.10.T
9.20.T	•	•			•	•	•	•	•	•		
9.14.T	•	•	•				•	•	•	•	•	•
9.12.T		•	•				•	•	•	•	•	•
8.21.T				•			•	•	•	•		
7.31.T	•	•	•		•	•	•	•	•	•	•	•
7.30.T	•	•	•		•	•	•	•	•	•	•	•
7.24.T	•	•	•	•	•	•	•	•	•	•	•	•
7.23.T	•	•	•	•	•	•	•	•	•	•	•	•
7.22.T	•	•	•	•	•	•	•	•	•	•	•	•
7.20.T	•	•			•	•	•	•	•	•		
7.12.T										•	•	
7.10.T											•	•



Database Server Migration Paths

This section summarizes the migration paths for Informix database servers. [Figure 1-4 on page 1-9](#) shows where you can find information in this book on how to upgrade to another Informix database server. [Figure 1-5 on page 1-10](#) shows where you can find information in this book on how to revert to a previous Informix database server. The **Source** column shows the original database server and environment. The **Target** column shows the new database server and environment to which you are migrating. The **Page Reference** column shows where you can find more information about the specific migration path.



Important: To update your database server, you must first perform a clean shutdown. Wait for all activity to cease before you proceed. Perform a level-0 backup as soon as the new database server is on-line.

[Figure 1-4](#) through [Figure 1-6](#), and other figures in this guide, use the following abbreviations for the database server names.

Database Server Name	Abbreviation
Informix Dynamic Server 2000	IDS 2000
INFORMIX-Universal Server	IUS
Informix Extended Parallel Server	XPS
Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options	AD/XP
INFORMIX-OnLine XPS	XPS
Informix Dynamic Server 7.3	IDS 7.3
Informix Dynamic Server, Workgroup and Developer Editions	W/D
Informix Dynamic Server, Linux Edition	LE
INFORMIX-OnLine Dynamic Server	ODS
INFORMIX-OnLine Workgroup Server	OWS
INFORMIX-SE	SE
INFORMIX-OnLine	OL

Figure 1-4
Upgrading Database Servers

Source	Target	Page Reference
IUS 9.14 on UNIX or Windows NT	IDS 2000 on UNIX or Windows NT	4-1
AD/XP 8.21 on UNIX or Windows NT	XPS 8.3 on UNIX	7-1
XPS 8.11 on UNIX	AD/XP 8.21 on UNIX or Windows NT	9-1
IDS 7.3x on UNIX or Windows NT	IDS 2000 on UNIX or Windows NT	4-1
	IUS 9.14 on UNIX or Windows NT	6-1
	XPS 8.3 on UNIX	8-1
	Later IDS 7.3x on UNIX or Windows NT	11-1
LE 7.3x on Linux	IDS 2000 on UNIX or Windows NT	4-1
ODS 7.2x on UNIX or Windows NT	IDS 2000 on UNIX or Windows NT	4-1
	IUS 9.14 on UNIX or Windows NT	6-1
	XPS 8.3 on UNIX	8-1
	AD/XP 8.21 on UNIX or Windows NT	10-1
	IDS 7.3x on UNIX or Windows NT	11-1
	Later ODS 7.2x on UNIX or Windows NT	11-1
ODS 6.0 to 7.1x on UNIX	IDS 7.3x on UNIX	11-1
ODS 7.12 on Windows NT	IDS 7.3x on Windows NT	11-1
OL 4.1 or 5.0x on UNIX	IDS 7.3x on UNIX	12-1
OWS 7.12 on UNIX	W/D 7.3x on UNIX	14-1
OWS 7.12 on Windows NT	OWS 7.22 on Windows NT	14-1
SE 7.2x or earlier	Later SE 7.2x	13-1

Figure 1-5
Reverting Database Servers

Source	Target	Page Reference
IDS 2000 on UNIX or Windows NT	IUS 9.14 on UNIX or Windows NT IDS 7.3x on UNIX or Windows NT ODS 7.2x on UNIX or Windows NT	4-1 4-1 4-1
XPS 8.3 on UNIX	AD/XP 8.21 on UNIX or Windows NT	7-1 8-1
AD/XP 8.21 on UNIX or Windows NT	XPS 8.11 on UNIX IDS 7.3x or ODS 7.2 x on UNIX or Windows NT	9-1 10-1
IUS 9.14 on UNIX or Windows NT	Earlier IUS 9.1x on UNIX or Windows NT IDS 7.3x on UNIX or Windows NT ODS 7.2x on UNIX or Windows NT	5-1 6-1 6-1
IDS 7.31 on UNIX or Windows NT	IDS 7.30 on UNIX or Windows NT ODS 7.2x to 6.0 on UNIX or Windows NT OL 5.0x or 4.1 on UNIX	11-1 12-1
IDS 7.30 on UNIX or Windows NT	ODS 7.2x to 6.0 on UNIX or Windows NT OL 5.0x or 4.1 on UNIX	11-1 12-1
ODS 7.22 on Windows NT	ODS 7.2x or 7.12 on Windows NT	11-1
OWS 7.22 on UNIX	OWS 7.12 on UNIX	14-1
W/D 7.3x on Windows NT	OWS 7.12 on Windows NT	14-1
SE 7.2x	Earlier SE 7.2x or earlier version	13-1

Migrating Between Operating Systems

Figure 1-6 shows the paths for migrating to a different operating system. For details, see [Chapter 15, “Moving a Database Server to a Different Operating System,”](#) or [Chapter 16, “Migrating Between Dynamic Server 7.3 and Its Editions in Different Environments.”](#)

Figure 1-6
Migrating to a Database Server in a Different Operating System

Source	Target
IDS 2000 on UNIX	IDS 2000 on Windows NT
IDS 2000 on Windows NT	IDS 2000 on UNIX
IUS on UNIX	IUS on Windows NT
IUS on Windows NT	IUS on UNIX
AD/XP on UNIX	AD/XP on Windows NT
AD/XP on Windows NT	AD/XP on UNIX
IDS 7.3 on UNIX	IDS 7.3, ODS, OWS, SE, or W/D on Windows NT
IDS 7.3 on Windows NT	IDS 7.3, ODS, OWS, SE, or W/D on UNIX
ODS on UNIX	IDS 7.3, OWS, SE, or W/D on Windows NT
ODS on Windows NT	IDS 7.3, OWS, SE, or W/D on UNIX
OWS or W/D on UNIX	IDS 7.3, ODS, OWS, SE, or W/D on Windows NT
OWS or W/D on Windows NT	IDS 7.3, ODS, OWS, SE, or W/D on UNIX
SE on UNIX	IDS 7.3, ODS, OWS, or W/D on Windows NT
SE on Windows NT	IDS 7.3, ODS, OWS, or W/D on UNIX
C-ISAM	IDS 7.3, ODS, SE, OWS, or W/D

Data Migration

In This Chapter	2-3
Migrating a Database or Selected Data	2-3
Migrating Between Different Versions of a Database Server	2-4
Changing Database Servers, Operating Systems, or GLS Locales	2-4
Distributing a Client Application	2-4
Importing Non-Informix Data.	2-5
Setting Environment Variables Before Using Utilities	2-5
Choosing the Most Effective Data Migration Tools	2-6
Automatic Data Migration	2-9
The onunload and onload Utilities	2-9
Constraints on Using onunload and onload	2-11
Restrictions on Using onunload and onload	2-12
The dbexport and dbimport Utilities	2-13
Destination Options	2-14
Location Options	2-14
Database Server Options	2-15
SE Options	2-15
LOAD, UNLOAD, and dbload	2-15
The dbschema Utility.	2-17
Guidelines for Using dbschema	2-17
DB-Access Input from dbschema Output	2-18
Object Modes and Violation Detection	2-18
External Tables	2-19
The High-Performance Loader on UNIX	2-19
HPL Tools	2-20
Performance Advantage of the HPL	2-20
The onxfer Utility	2-20
Informix Enterprise Command Center.	2-21

Moving Data Between Computers and Dbspaces	2-21
Importing Data from a Non-Informix Source.	2-21
Importing Data with Informix Enterprise Gateway Products	2-22

In This Chapter

This chapter provides an overview of data migration and compares the Informix migration utilities. This chapter covers the following topics:

- Migrating a database or selected data
- Setting environment variables before using utilities
- Choosing the most effective data migration tool
- Moving data between computers and dbspaces

Migrating a Database or Selected Data

You might need to perform data migration on a database or selected data to complete any of the following tasks:

- Migrating between different versions of a database server
- Changing database servers, operating systems, or GLS locales
- Distributing a client application
- Importing non-Informix data

Migrating Between Different Versions of a Database Server

When you upgrade to a newer version or revert to a previous version of a database server, you need to consider the following data migration issues:

- Changes in the configuration parameters and environment variables
- Amount of memory and dbspace space required
- Organization of the data

Chapters in Sections II, III, IV, and V discuss these issues in detail.

Changing Database Servers, Operating Systems, or GLS Locales

You might want to move between database servers, to a different operating system, or to a different GLS locale. You also might want to change the database schema to accommodate more information, to provide for growth, or to enhance performance.

If you are migrating to Dynamic Server with AD and XP Options 8.21 in a different operating system or to SE, you must unload the data into text files and then load it back again into the target database server. ♦

For information about moving data to a different operating system, see Section V, [“Migration Between Operating Systems.”](#)

For information about moving to a GLS locale, see [Chapter 18, “Converting to GLS.”](#) ♦

Distributing a Client Application

After you upgrade a database server on the same operating system or move the database server to another, compatible computer, review the client applications and **sqlhosts** file or registry-key connections. You might need to recompile or modify client applications, or update the **sqlhosts** file or registry key.

AD/XP

SE

GLS

Verify that the client-application version you use is compatible with your database server version. Update the **sqlhosts** file or registry key for the client applications with the new database server information.

For more information about interactions between client applications and different database servers, refer to a client manual, such as the *Informix ESQL/C Programmer's Manual* or the *DataBlade API Programmer's Manual*.

Importing Non-Informix Data

You can use the **dbimport** and **dbload** utilities, the High-Performance Loader (HPL), the Informix Enterprise Gateway products, or external tables to import data from non-Informix sources.

Setting Environment Variables Before Using Utilities

Before you use any data migration utility, you must set your **PATH**, **INFORMIXDIR**, and **INFORMIXSERVER** environment variables. For information about environment variables, see the *Informix Guide to SQL: Reference*.

***Tip:** If you are using SE, Informix recommends that you not use the **.dbs** directory as your current directory when you use a database-related utility. This practice keeps the **.dbs** directory free from file clutter and prevents multiple users from overwriting files that belong to other users.*

For information about SE administration utilities, see your *INFORMIX-SE Administrator's Guide*. ♦



SE

Choosing the Most Effective Data Migration Tools

You can use the following tools, utilities, and SQL statements to move data from one database to another:

- The **onunload** and **onload** utilities (only between database servers of the same version)
- The **dbexport** and **dbimport** utilities
- UNLOAD and LOAD statements and the **dbload** utility
- The **dbschema** utility
- External tables
- The High-Performance Loader (HPL)
- The **onxfer** utility
- Informix Enterprise Command Center (IECC), for Version 8.21, Version 7.3x, and some earlier database servers

This section provides brief descriptions of these data migration tools and information to help you choose the most effective ones to move your data. For additional details about using these tools and for utility command and statement syntax, see the chapters in Section VII, “[Data Migration Utilities](#).” For more information about external tables and the HPL, see the *Administrator’s Reference*, your *Administrator’s Guide*, or the *Informix Guide to SQL: Syntax*. For more information about IECC, see the *Informix Enterprise Command Center User Guide*.

[Figure 2-1 on page 2-7](#) lists which tools you can use for each database server.

Figure 2-1
Utilities for Moving Data

Utility	Database Server								
	IDS 2000	IUS	XPS (8.3)	AD/XP	XPS (8.11)	IDS 7.3	OWS or W/D	ODS	OL
dbexport/dbimport	•	•				•	•	•	•
dbload	•	•				•	•	•	•
External tables			•	•	•				
HPL	•	•				•		•	
IECC				•		•	•	•	
onunload/onload	•	•				•	•	•	
onxfer			•	•					
UNLOAD/LOAD	•	•	•	•	•	•	•	•	•

The best method for moving data depends on your operating system and whether you want to move an entire database, selected tables, or selected columns from a table. [Figure 2-2 on page 2-8](#) summarizes the characteristics of the methods for loading data and the constraints and advantages of each method.

Figure 2-2
Comparison of Tools for Loading Data

	dbexport/ dbimport	dbload	External Tables	HPL	IECC	onunload/ onload	onxfer	UNLOAD/ LOAD
Granularity of Data	Database only	Partial or complete table	Partial or complete table	Partial or complete table	Table or database	Table or database	Table or database	Partial or complete table
Performance	Moderate	Slow	Very fast	Fast	Moderate	Fast	Very fast	Moderate
Source of Data	Usually produced by dbexport	Any data in the format specified by the input file	Any data in the format specified by the input file	Any ASCII or COBOL data. User can create custom read capabilities.	Data must be unloaded by IECC	Must be produced by onunload	Table or database	Any data in the specified format, usually produced by UNLOAD
Database Schema	Can modify	Can modify	Can modify	Can modify	Cannot modify	Cannot modify	Can modify	Can modify
Location of Data	Disk or tape	Disk only	Disk, tape, or pipe	Disk, tape, or pipe	Disk or tape	Disk or tape	Disk or pipe	Disk only
Type of File	Text	Text	Text	Text	Text	Binary	Text	Text
Logging Status	Logging optional	Logging optional	Logging optional	Logging optional	Logging optional	Logging must be turned off	Logging optional	Logging optional
Operating System	Can use when you move data between operating systems	Can use when you move data between operating systems	Can use when you move data between operating systems or from a non-Informix database	Can use when you move data between operating systems or from a non-Informix database	Can use when you move data between operating systems	Cannot use when you move data between operating systems	Can use when you move data between operating systems	Can use when you move between operating systems
Ease of Use	Moderate	Moderate	Most difficult	Most difficult	Easy to use	More difficult	Moderate	Easiest

The following subsections provide guidelines on how to choose the appropriate migration tool or tools.

Automatic Data Migration

Figure 2-3 shows situations where data migration is automatic. Automatic data migration means that when you migrate from one database server to another, you do not need to use any data migration tools to move the data. The data migrates automatically from the source database server to the target database server after you bring up the target database server. For example, data migration between Dynamic Server 7.3 and OnLine Workgroup Server is automatic if the database servers use the same operating system.

Data migration can also be automatic when you move between different versions of a database server in the same operating system.

Source	Target	Page Reference
ODS, IDS 7.3	OWS, W/D	14-1
OWS, W/D	ODS, IDS 7.3	14-1

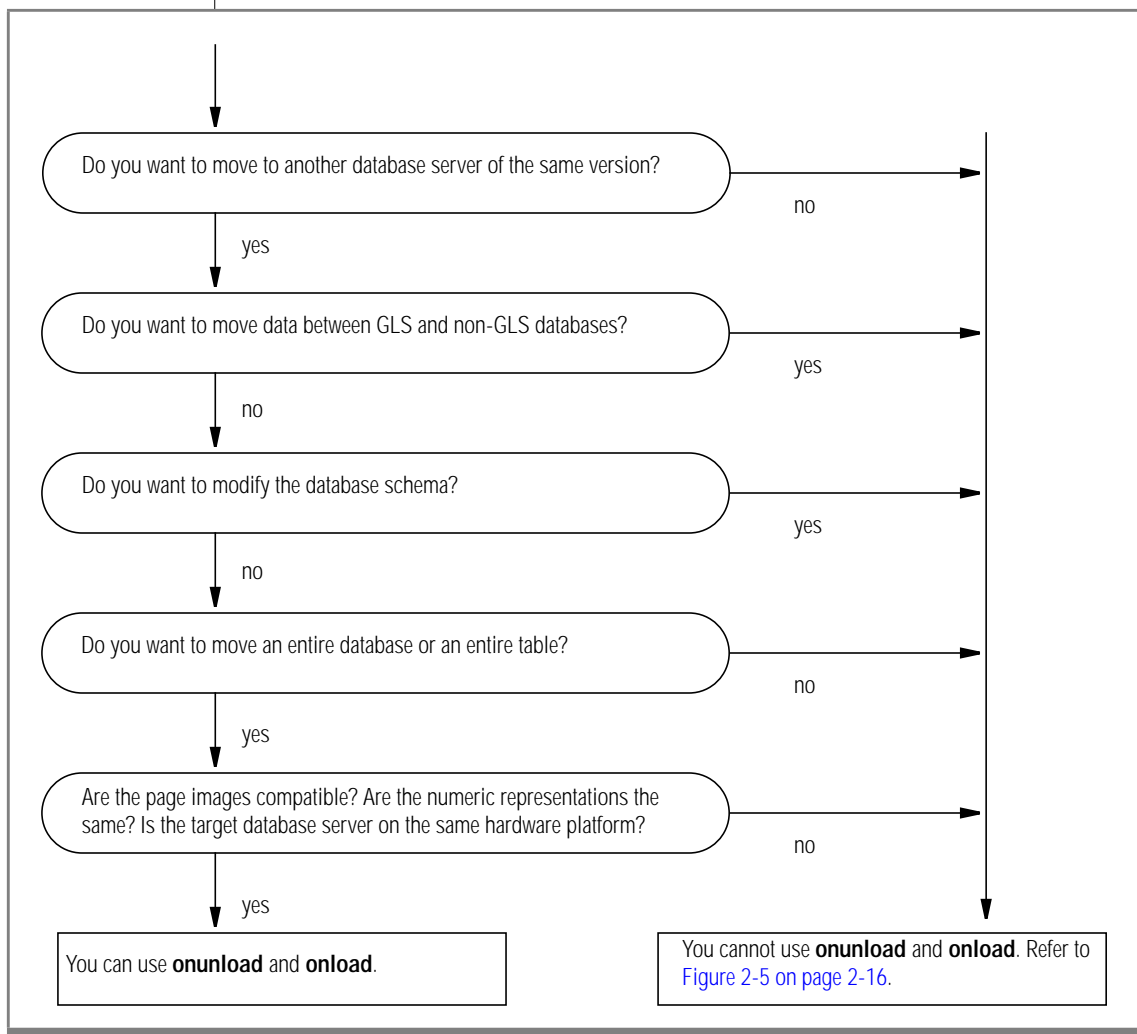
Figure 2-3
*Migrating Data Automatically
Between Different Database
Servers*

The onunload and onload Utilities

The **onunload** and **onload** utilities provide the fastest way to move data, but they do not let you modify the database schema or move from one operating system or database server version to another. The **onunload** utility unloads data from the specified database or table onto a tape or a file on disk in disk-page-sized units, making this utility more efficient than **dbexport**. The **onload** utility takes a tape or a file that the **onunload** utility creates and re-creates the database or the table. The **onunload** and **onload** utilities are faster than **dbimport**, **dbload**, or **LOAD** but are much less flexible.

Because the data is written in page-sized units, you can use **onunload** and **onload** only when certain constraints are met. For example, you cannot use **onunload** and **onload** to move data between UNIX and Windows NT. You can, however, use the **onunload** and **onload** utilities to move data between computers. [Figure 2-4](#) summarizes the questions that you must ask before you use **onunload** and **onload**.

Figure 2-4
*Criteria for Using the **onunload** and **onload** Utilities*



Constraints on Using *onunload* and *onload*

The **onunload** and **onload** utilities are the fastest way to unload and load data, but you can use them only when all the following criteria are the same for the source and target computers:

- Page size
- Representation of numeric data
- Byte alignment for structures and unions
- Informix database server version

You cannot use **onunload** and **onload** to move data between UNIX and Windows NT because they use different page sizes. For example, the page size is 2 kilobytes on some UNIX systems and 4 kilobytes on Windows NT.

You can use the **onunload** and **onload** utilities to unload from and load data into the following database servers between operating systems.

Database Server	Operating System(s)
Dynamic Server 2000	UNIX or Windows NT (only if the database does not contain extended data types)
Universal Server	UNIX or Windows NT (only if the database does not contain extended data types)
Dynamic Server 7.3	UNIX or Windows NT
Dynamic Server, Workgroup and Developer Editions	UNIX or Windows NT
OnLine Dynamic Server	UNIX or Windows NT
OnLine Workgroup Server	UNIX or Windows NT

For example, your site purchases a more powerful UNIX computer to allow faster access for users. You need to transfer existing databases to the new database server on the new computer. Use **onunload** to unload data from the first database server and then use **onload** to load the data into the second database server. Both database servers must have the same version number, or they must have compatible version numbers. You can move the entire database or selected tables only, but you cannot modify the database schema.

Restrictions on Using onunload and onload

The **onunload** and **onload** utilities have the following restrictions:

- You cannot use **onunload** and **onload** to move data between non-GLS and GLS locales. ♦
- Do not use **onunload** and **onload** to move data between two Dynamic Server 2000 or Universal Server databases if either database contains an extended data type.

Use the HPL instead to move the data. However, you can use **onunload** and **onload** with Dynamic Server 2000 or Universal Server if the database contains only legacy data types. ♦

- SE does not support **onunload** and **onload**. However, you can use **dbexport** and **dbimport**. ♦
- Dynamic Server with AD and XP Options, Extended Parallel Server, and OnLine XPS do not support **onunload** and **onload**.

For Dynamic Server with AD and XP Options or OnLine XPS, use external tables to unload and load your data or to move data between operating systems. For more information on how to unload and load this data, see [“External Tables” on page 2-19](#) and [“Loading and Unloading Data” on page 10-19](#). ♦

You can use **onunload** and **onload** to move data between databases if the NLS and GLS locales are identical. For example, if user A has a French locale NLS table on server A and tries to load data into a German locale GLS table on server B, **onload** and **onunload** report errors. However, if both the NLS and GLS tables were created with the same French locale, **onload** and **onunload** would work.

GLS

IDS 2000

IUS

SE

AD/XP

XPS 8.3

XPS 8.11

The tape that **onload** reads contains binary data that is stored in disk-page-sized units. For this reason, the computers where the original database resides (where you use **onunload**) and where the target database will reside (where you use **onload**) must have the following characteristics:

- The same page size
- The same representation of numeric data
- The same byte alignment for structures and unions

If the page sizes are different, **onload** fails. If the alignment or numeric data types on the two computers are different (for example, with the most-significant byte last instead of first or different float-type representations), the contents of the data page could be misinterpreted.



***Important:** You cannot use the **onload** and **onunload** utilities to move data from one version of a database server to another. You also cannot use these utilities to move data between different types of database servers.*

For additional constraints and restrictions, see [Chapter 25, “Using the **onunload** and **onload** Utilities.”](#)

The **dbexport** and **dbimport** Utilities

If you cannot use **onunload** and **onload**, you must choose among the **dbload** utility (to load), the **dbexport** and **dbimport** utilities, and the UNLOAD and LOAD SQL statements. All these methods enable you to modify the database schema. The **dbexport** and **dbimport** utilities provide some flexibility, but you must move an entire database.

If you cannot use **onunload** and **onload** to export and import data, you can unload your data to text files. You can use the **dbexport** utility to unload data to tape from any of the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- SE
- Dynamic Server, Workgroup and Developer Editions

The UNLOAD statement lets you manipulate the data as you unload it, but it requires that you unload to files on disk instead of to tape. If you unload to files, you might need to use UNIX or Windows NT utilities to load those files onto tape.

The **dbexport** utility unloads a database into text files and creates a schema file. You can use the schema file with **dbimport** to re-create the database schema in another Informix environment. You can edit the schema file to modify the database that **dbimport** creates. The **dbexport** utility supports Dynamic Server 2000 and Universal Server data types. ♦

Destination Options

The **dbexport** utility supports the following destination options:

- Unload a database and its schema file to disk
- Unload a database and its schema file to tape
- Unload the schema file to disk and unload the data to tape

The **dbimport** utility creates a database and loads it with data from text files. The input files consist of a schema file that is used to re-create the database and data files that contain the database data. Normally, you generate the input files with the **dbexport** utility, but you can use any properly formatted input files. The **dbimport** utility supports new data types in Dynamic Server 2000 and Universal Server.

Location Options

The **dbimport** utility can use files from the following location options:

- All input files are located on disk
- All input files are located on tape
- The schema file is located on disk, and the data files are located on tape

Database Server Options

The **dbimport** utility supports the following options for a new Informix database server (except SE):

- Create an ANSI-compliant database (includes unbuffered logging).
- Establish transaction logging for a database (unbuffered or buffered logging).
- Specify the dbspace where the database will reside.

SE Options

The **dbimport** utility supports the following options for a new SE database:

- Create an ANSI-compliant database (ANSI-compliant logging).
- Establish transaction logging for a database (unbuffered logging).

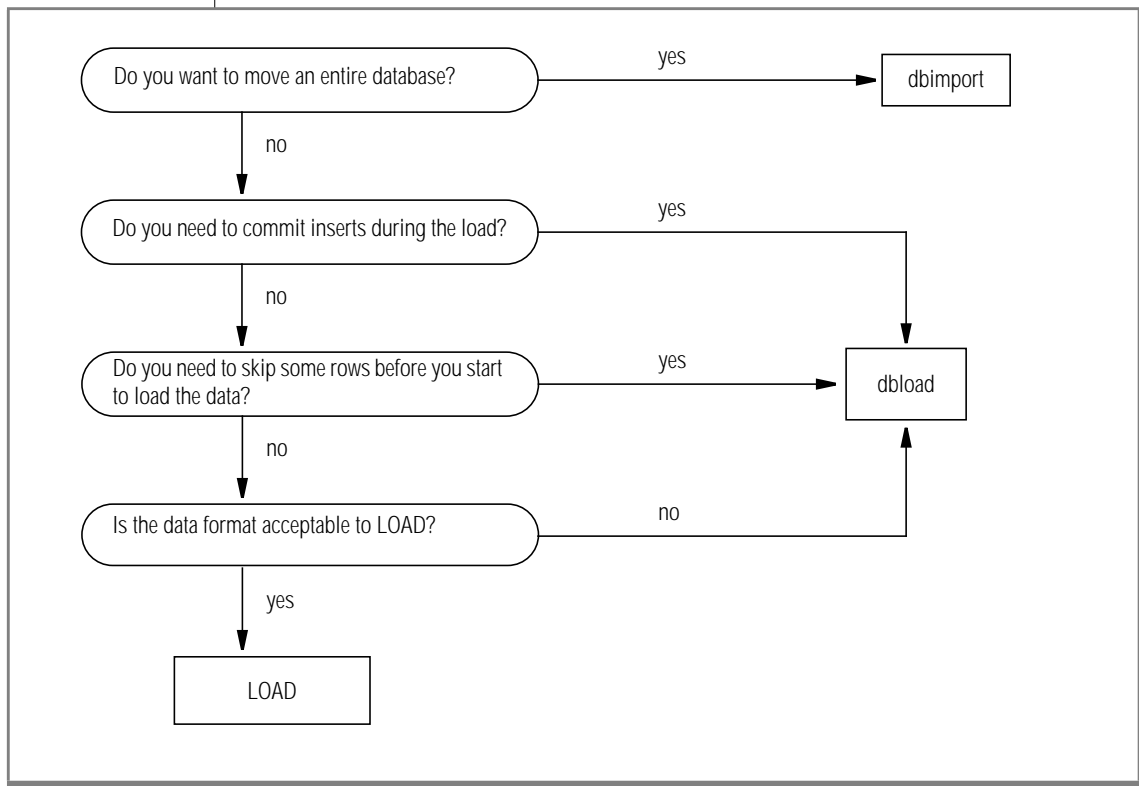
LOAD, UNLOAD, and dbload

The LOAD statement is moderately fast and easy to use, but it can only accept specified data formats. You usually use LOAD with data that is prepared with an UNLOAD statement.

You can use the UNLOAD statement in DB-Access to unload selected rows from a table into a text file.

To load tables, use LOAD or **dbload**. To manipulate a data file that you are loading or to access a database while it is loading, use the **dbload** utility. The cost of the flexibility is the time you spend creating the **dbload** command file and slower execution. When possible, use the LOAD statement, which is faster than **dbload**. [Figure 2-5](#) summarizes questions to help you choose among these methods.

Figure 2-5
Choosing Among dbimport, dbload, and LOAD



The **dbload** utility gives you a great deal of flexibility, but it is not as fast as the other methods, and you must prepare a command file to control the input. You can use **dbload** with data in a variety of formats.

The **dbload** utility offers the following advantages over the LOAD statement:

- You can use **dbload** to load data from input files that were created with a variety of format arrangements. The **dbload** command file can accommodate data from entirely different database management systems.
- You can specify a starting point in the load by directing **dbload** to read but ignore *x* number of rows.
- You can specify a batch size so that after every *x* number of rows are inserted, the insert is committed.
- You can limit the number of bad rows read, beyond which **dbload** ends.

The cost of **dbload** flexibility is the time and effort spent creating the **dbload** command file, which is required for **dbload** operation. The input files are not specified as part of the **dbload** command line, and neither are the tables into which the data is inserted. This information is contained in the command file.

The *dbschema* Utility

You can use the **dbschema** utility for the following purposes:

- To display the SQL statements (the *schema*) that are required to replicate a database or a specific table, view, or procedure
- To display the schema for the Information Schema views
- To display the distribution information that is stored for one or more tables in the database
- To display information on user-defined data types and row types

Guidelines for Using dbschema

For Dynamic Server with AD and XP Options, Extended Parallel Server, or OnLine XPS, you need to edit your schema files to incorporate the *dbslice* feature and the syntax definition of *dbspaces*. In these database servers, the **dbschema** utility produces *dbslice* information. For information on these schema issues, see Section III, [“Version 8.x Database Server Migration.”](#) ♦

AD/XP

XPS 8.3

XPS 8.11

SE

GLS

If you are using SE, the database must exist in your current directory or in a directory that is cited in your **DBPATH** environment variable. ♦

When the GLS environment variables are set correctly, as the *Informix Guide to GLS Functionality* describes, **dbschema** can handle foreign characters in Dynamic Server 2000, Dynamic Server 7.3, Universal Server, or OnLine Dynamic Server 7.2 databases. ♦

You can use delimited identifiers with the **dbschema** utility. The utility detects database objects that are keywords, mixed case, or have special characters, and places double quotes around them.

DB-Access Input from dbschema Output

You can use the **dbschema** utility to get the schema of a database and redirect the **dbschema** output to a file. Later, you can feed this file to DB-Access to re-create the database.

Object Modes and Violation Detection

The **dbschema** output supports object modes and violation detection, as follows:

- The output shows the names of not-null constraints after the not-null specifications.
You can use the output of the utility as input to create another database. If the same names were not used for not-null constraints in both databases, problems could result.
- The output shows the object mode of objects that are in the disabled state. These objects can be constraints, triggers, or indexes.
- The output shows the object mode of objects that are in the filtering state. These objects can be constraints or unique indexes.
- The output shows the violations and diagnostics tables that are associated with a base table (if violations and diagnostics tables were started for the base table).

For more information about object modes and violation detection, see the SET, START VIOLATIONS TABLE, and STOP VIOLATIONS TABLE statements in the *Informix Guide to SQL: Syntax*.

AD/XP

XPS 8.11



UNIX

External Tables

You can use external tables to unload and load data when no other data migration tool is available.

For Dynamic Server with AD and XP Options or OnLine XPS, you need to use external tables to unload and load data. These database servers use high-performance parallel loading with external tables. External tables can hold data that you back up from a database server. You can move the data from the external tables to another database server.

To load and unload data in Dynamic Server with AD and XP Options or in OnLine XPS, use external tables formatted in the Informix internal data representation format. You can load and unload files with the default delimiter (|) format. ♦

Important: *You do not need to use external tables to move data to Extended Parallel Server from Dynamic Server 7.3 or OnLine Dynamic Server 7.2. You can use the **onxfer** utility instead. For information on using the **onxfer** utility, see [Chapter 8, “Migrating Between Extended Parallel Server and a 7.3 or 7.2 Database Server.”](#)*

For detailed information on external tables, see the *Administrator’s Reference* or your *Administrator’s Guide*. For syntax, see the *Informix Guide to SQL: Syntax*.

The High-Performance Loader on UNIX

The High-Performance Loader (HPL) uses parallel processing to perform fast data loading and unloading. The HPL utility is available with the following database servers:

- Dynamic Server 2000
- Dynamic Server 7.3
- Universal Server
- OnLine Dynamic Server 7.2

Only the UNIX versions of these database servers support the HPL.

The HPL requires significant preparation time but is fast. Use the HPL for large migration jobs. The HPL can load data from any ASCII or COBOL file that meets certain format requirements.

You can use the HPL to load from large ASCII or COBOL databases. COBOL is supported up to Version 7.3.

HPL Tools

In addition to the advantage of speed, the following HPL features provide powerful tools for handling data from non-Informix sources:

- Drivers to handle different database types
- Filters and functions to manipulate data
- Code-set conversion

The **onpload** utility is the command-line portion of the HPL.

Performance Advantage of the HPL

For extremely large databases, the HPL has a performance advantage over other Informix data-migration utilities because it performs I/O and code-set conversions in parallel. The user, however, must invest significant preparation time before using the HPL, and the HPL program has a significant start-up time. Therefore, use the HPL only for large databases, for which the time savings in the actual loading or unloading of data makes the preparation time worthwhile.

For more information about the HPL, refer to the *Guide to the High-Performance Loader*.

XPS 8.3

The onxfer Utility

For Extended Parallel Server, you can use the **onxfer** utility to move a database or individual tables to Extended Parallel Server coservers from Dynamic Server 7.3 or OnLine Dynamic Server 7.2. You can use **onxfer** to transfer a database or selected tables to disk. This data movement utility combines the speed of the HPL and external tables functionality with the usability of the **dbexport** and **dbimport** utilities.

Informix Enterprise Command Center

Informix Enterprise Command Center (IECC) provides a graphical user interface that you can use on 8.2x, 7.3x, and 7.2x database servers to load and unload single tables or entire databases to and from Informix database servers. For more information, refer to the *Informix Enterprise Command Center User Guide*.

Moving Data Between Computers and Dbspaces

This section discusses moving data between different computers and importing data from non-Informix environments. Except when you use the HPL or external tables, you must unload your data to ASCII files before you move the data to another computer.

If you are moving to an Informix database server on another computer, you can use the **dbimport** and **dbload** utilities to load the data that you exported.

If you are moving data to a non-Informix application, you might need to use the UNLOAD statement because it lets you specify the delimiter that is used in the data files.

Importing Data from a Non-Informix Source

The **dbimport** and **dbload** utilities can import data from any ASCII file that is properly formatted. Most applications that produce data can export the data into files that have a suitable format for **dbimport**. If the format of the data is not suitable, use UNIX or Windows NT utilities to reformat the data before you import it into one of the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE

In addition to **dbimport** and **dbload**, the Informix Enterprise Gateway products and the HPL provide ways to access information from non-Informix sources.

Importing Data with Informix Enterprise Gateway Products

Informix Enterprise Gateway with DRDA lets you query databases that conform to the DRDA protocol published by IBM. You can use this Gateway product to query a DRDA database and then insert the results into an Informix database. For example, to import data, execute a SELECT statement from the non-Informix database and then an INSERT statement into the Informix database. For more information, refer to the *Informix Enterprise Gateway with DRDA User Manual*.

Informix Enterprise Gateway for EDA/SQL lets you issue queries on a variety of hardware platforms. It accesses an EDA/SQL database server from Information Builders, Inc., which in turn accesses the data source. For more information, refer to the *INFORMIX-Enterprise Gateway for EDA/SQL User Manual*.

UNIX

Informix Enterprise Gateway provides a single, standards-based gateway to multiple data sources. Gateway Manager connects the Informix environment with that of any shared-library ODBC Level 2-compliant driver manager and driver(s) on UNIX. For instance, you can use Gateway Manager with the Informix Enterprise Gateway driver products to access UNIX database server products such as SYBASE SQL Server 10 and ORACLE7 Server. For more information, refer to the *INFORMIX-Enterprise Gateway Manager User Manual*. ♦

Client Software Development Kit (SDK) Compatibility

In This Chapter 3-3

In This Chapter

This chapter provides information about the compatibility of Informix Client Software Developer's Kit (Client SDK) with Informix database servers.

[Figure 3-1 on page 3-4](#) shows which versions of the Client SDK you can use with which database servers.

Figure 3-1
Client SDK Compatibility with Database Servers

	Client SDK 2.30	Client SDK 2.20	Client SDK 2.10	Client SDK 2.01
Dynamic Server 2000	•	•		
Universal Server		•	•	
Extended Parallel Server	•	•		
Dynamic Server with AD and XP Options		•	•	
OnLine XPS		•	•	•
Dynamic Server 7.3		•	•	•
Dynamic Server, Workgroup and Developer Editions		•	•	•
OnLine Dynamic Server			•	•
OnLine Workgroup Server			•	•
SE			•	•
OnLine			•	•

Version 9.x Database Server Migration

- Chapter 4** **Migrating Between Dynamic Server 2000 and a 9.14, 7.3, or 7.2 Database Server**
- Chapter 5** **Migrating Between Versions of Universal Server**
- Chapter 6** **Migrating Between Universal Server and OnLine Dynamic Server 7.2x or 7.1x**

Section II



Migrating Between Dynamic Server 2000 and a 9.14, 7.3, or 7.2 Database Server

In This Chapter	4-3
Preparing for Migration	4-4
Migration Guidelines.	4-4
Changes That Dynamic Server 2000 Introduced	4-5
Environment Variables	4-5
Configuration Parameters.	4-6
SQL Reserved Words	4-6
Migration from Universal Server 9.10 to 9.13	4-7
Migration from OnLine 4.1 or 5.0 or OnLine Dynamic Server 6.0 or 7.1	4-8
Storage-Manager Validation and Installation	4-8
Upgrading to Dynamic Server 2000 from 9.14, 7.3x, or 7.2x	4-9
Check and Configure Available Space	4-10
Save Copies of the Current Configuration Files.	4-12
Stop Enterprise Replication	4-13
Close All Transactions and Shut Down the Source Database Server	4-14
Check for Any Open Transactions	4-14
Verify the Integrity of the Data	4-15
Verify the Database Server Mode.	4-16
Make a Final Backup of the Source Database Server	4-16
Verify That the Source Database Server Is Off-Line	4-17
Modify UNIX Kernel Parameters	4-17
Install Dynamic Server 2000	4-17
Set Environment Variables	4-18
Customize the ONCONFIG Configuration File.	4-19

Add Any Communications Support Modules	4-19
For ON-Archive, Update Configuration Files	4-20
Install and Configure Any Datablade Modules	4-20
Initialize Dynamic Server 2000	4-21
Monitor the Upgrade Complete Status	4-21
Update Statistics	4-22
Verify the Integrity of the Data	4-25
Make an Initial Backup of Dynamic Server 2000	4-25
Tune Dynamic Server 2000 for Performance	4-26
Reverting to Universal Server 9.14, Dynamic Server 7.3x,or	
OnLine Dynamic Server 7.2x from Dynamic Server 2000	4-26
Determine Whether Reversion Is Possible.	4-27
Save Copies of the Configuration Files	4-31
Stop Enterprise Replication.	4-31
Verify the Integrity of the Data	4-31
Back Up Dynamic Server 2000.	4-31
Remove Version 9.2 Features	4-32
Run the Reversion Utility	4-32
Modify Configuration Parameters	4-33
Reset Environment Variables	4-33
Remove Any Communications Support Module Settings	4-33
Reinstall and Start the Target Database Server	4-34
Update Statistics	4-34
Verify the Integrity of the Data	4-36
Back Up the Target Database Server.	4-36
Return the Target Database Server to On-Line Mode	4-36



In This Chapter

This chapter describes migration between Dynamic Server 2000 and Universal Server 9.14, Dynamic Server 7.3x (Version 7.30 or Version 7.31), or OnLine Dynamic Server 7.2x. This chapter covers the following topics:

- Preparing for migration
- Upgrading to Dynamic Server 2000 from Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x
- Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000

Important: To migrate to Dynamic Server 2000 from Universal Server 9.10, 9.11, 9.12, or 9.13, you must first upgrade to Universal Server 9.14. See [“Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13” on page 5-3.](#)

To migrate to Dynamic Server 2000 from a version of OnLine Dynamic Server earlier than Version 7.20, you must first upgrade to OnLine Dynamic Server 7.2x or Dynamic Server 7.3x. For more information, see [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#)

The instructions for Dynamic Server 7.3x also apply to Dynamic Server, Linux Edition, Version 7.30 or Version 7.31, and to Dynamic Server, Workgroup and Developer Editions, Version 7.30 or Version 7.31.

Preparing for Migration

To prepare for migration between Dynamic Server 2000 and Universal Server 9.14, Dynamic Server 7.3, or OnLine Dynamic Server 7.2, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration.

To prepare for migration from Dynamic Server 7.3x or from OnLine Dynamic Server 7.2x, see [“Changes That Universal Server Introduced” on page 6-5](#). In addition, for migration from OnLine Dynamic Server, see [“Environment Variable Changes in Dynamic Server 7.3” on page 11-6](#), [“New Configuration Parameters in Dynamic Server 7.3” on page 11-7](#), and [“New Features in Dynamic Server 7.3” on page 11-8](#).

To migrate from an earlier database server, you need to upgrade to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x before you upgrade to Dynamic Server 2000. To migrate from an earlier version of Universal Server, see [“Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13” on page 5-3](#). To migrate from an earlier version of OnLine Dynamic Server, see [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#)

Migration Guidelines

Informix suggests that you observe the following precautions when you migrate to Dynamic Server 2000:

- Check the release notes for information about the correct operating-system release and any patches that you need for successful installation and operation of the database server.

The release notes are in one of the following directories:

- `$INFORMIXDIR/release/en_us/0333`. ♦
- `%INFORMIXDIR%\release\en_us\0333`

Release notes are in the **Informix** folder. To display this folder, choose **Start→Programs→Informix** from the task bar. ♦

- On UNIX, retain both versions of the Informix product software on disk, if you have enough disk resources. You cannot retain both versions of the Informix product on disk on Windows NT.

UNIX

WIN NT

- Retain the installation media from both versions of the Informix product software.
- Before you upgrade to the target database server from the source database server, make sure that no open transactions exist in the source database server.

Fast recovery will fail when rolling back open transactions during the upgrade. For information about how to close the source database server properly prior to upgrading, see [“Close All Transactions and Shut Down the Source Database Server” on page 4-14](#).

- Before migration, perform a level-0 backup of all storage spaces with the source database server. After you complete the migration, perform another level-0 backup with Dynamic Server 2000.
- Use a test instance of your database server to test the installation and migration procedures.
- Verify storage-manager validation for the target database server. For details, see [“Storage-Manager Validation and Installation” on page 4-8](#).

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Changes That Dynamic Server 2000 Introduced

This section describes some of the changes in Dynamic Server 2000 that affect migration or initial configuration.

Environment Variables

Dynamic Server 2000 introduced the following new environment variables that were not available in Universal Server:

- **IFX_LONGID**
- **IFX_UPDESC**
- **STMT_CACHE**
- **STMT_CACHE DEBUG**

Review the descriptions of these environment variables in the *Informix Guide to SQL: Reference* to determine whether you need to set them.

Configuration Parameters

Dynamic Server 2000 introduced new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. The *Administrator's Reference* describes the following new configuration parameters:

- ALLOW_NEWLINE
- DD_HASHMAX
- DD_HASHSIZE
- OPT_GOAL
- PC_HASHSIZE
- PC_POOLSIZE
- STMT_CACHE
- STMT_CACHE_SIZE
- SYSSBSPACENAME

SQL Reserved Words

Dynamic Server 2000 supports new SQL keywords that are reserved words and might affect migration of your applications. Although you can use almost any word as an SQL identifier, syntactic ambiguities can occur. An ambiguous statement might not produce the results you want.

This section lists the new SQL reserved words. For more information about SQL reserved words, see the *Informix Guide to SQL: Syntax*.

The following SQL keywords are new in Dynamic Server 2000:

- CACHE
- COSTFUNC
- ITEM
- SELCONST

The following SQL keywords are new in Dynamic Server 2000 and Dynamic Server 7.31:

- INNER
- JOIN
- LEFT
- LOCKS
- RETAIN

The following SQL keywords are new in Dynamic Server 2000 and Dynamic Server 7.30:

- ALL_ROWS
- CASE
- CRCOLS
- DECODE
- FIRST_ROWS
- MEMORY_RESIDENT
- NON_RESIDENT
- NVL
- REPLICATION
- SUBSTR
- SUBSTRING

Migration from Universal Server 9.10 to 9.13

Before you upgrade to Dynamic Server 2000 from Universal Server 9.10, 9.11, 9.12, or 9.13, you need to upgrade to Universal Server 9.14. For more information, see [“Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13” on page 5-3.](#)

Migration from OnLine 4.1 or 5.0 or OnLine Dynamic Server 6.0 or 7.1

Before you upgrade to Dynamic Server 2000 from OnLine 4.1x or 5.0x or from OnLine Dynamic Server 6.0x or 7.1x, you need to upgrade to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x. For more information, see [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0”](#) or [Chapter 12, “Migrating Between a 7.x or 6.0 Database Server and a 5.x or 4.1 Database Server.”](#)

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the *ism_catalog* utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Upgrading to Dynamic Server 2000 from 9.14, 7.3x, or 7.2x

This section describes the procedures for upgrading to Dynamic Server 2000 from any of the following database servers:

- Universal Server 9.14
- Dynamic Server 7.3x
 - Dynamic Server, Version 7.30 or Version 7.31
 - Dynamic Server, Workgroup and Developer Editions, Version 7.30 or Version 7.31
 - Dynamic Server, Linux Edition, Version 7.30 or Version 7.31
- OnLine Dynamic Server 7.2x

When you migrate to Dynamic Server 2000, you can install and test a database server instance with the same configuration files, environment variables, and **sqlhosts** information that you used for your source database server. After you install Dynamic Server 2000 and verify that it works, you might want to modify configuration files and environment variables to take advantage of Dynamic Server 2000 features. For more information, refer to your *Getting Started* manual and your *Administrator's Guide*.

To upgrade to Dynamic Server 2000, you need to complete the following steps. Later sections describe these steps in detail.

1. Check and configure available space.
2. Save copies of the current configuration files.
3. Stop Enterprise Replication.
4. Close all transactions and shut down the source database server.
5. Check for any open transactions.
6. Verify the integrity of the data.
7. Verify the database server mode.
8. Make a final backup of the source database server.
9. Verify that the source database server is off-line.

Important: Repeat steps 2 through 9 for each instance of an earlier database server that you are upgrading to Dynamic Server 2000.



UNIX



10. Modify UNIX kernel parameters. ♦
11. Install Dynamic Server 2000.

Important: Monitor the message log, **online.log**, during the upgrade for any error messages.

12. Set environment variables.
13. Customize the ONCONFIG configuration file.
14. Add any Communications Support Modules.
15. For ON-Archive, update configuration files.
16. Install and configure any DataBlade modules.
17. Initialize Dynamic Server 2000.
18. Monitor the upgrade complete status.
19. Update statistics.
20. Verify the integrity of the data.
21. Make an initial backup of Dynamic Server 2000.
22. Tune Dynamic Server 2000 for performance

Important: Repeat steps 13 through 22 for each instance of Dynamic Server 2000 that you plan to run on the computer.



Warning: If a serious error occurs during the upgrade, it might be necessary to return to the previous version, restore from tape, and then correct the problem prior to restarting the update procedure at Step 9.

Check and Configure Available Space

UNIX

Dynamic Server 2000 requires 3000 free pages of logical-log space (approximately 6000 kilobytes for a 2-kilobyte page size) to build the **sysmaster** database on UNIX. ♦

WIN NT

Dynamic Server 2000 requires 1500 to 3000 free pages of logical-log space (approximately 6000 kilobytes for a 4-kilobyte page size) to build the **sysmaster** database on Windows NT. ♦

You need to add any additional free space to the system prior to the upgrade. If the dbspaces are full, you need to add space before you start the upgrade procedure.

When you initialize Dynamic Server 2000 on an existing Universal Server root dbspace, the database server automatically upgrades the **sysmaster** database and then each database individually. For a successful upgrade of each database, you must ensure that 2000 kilobytes of free space per database is available in each dbspace, as follows:

1. Calculate the amount of free space that each dbspace requires. In the following equation, *n* is the number of databases in the dbspace and *X* is the kilobytes of free space they require:

$$X \text{ kilobytes free space} = 2000 \text{ kilobytes} * n$$

2. Check the amount of free space in each dbspace to determine whether you need to add more space.

Use the following SQL statements to determine the free space that each dbspace requires and the free space available. These statements return the free-space calculation in page-size units. The **free_space_req** column value is the free-space requirement, and the **free_space_avail** column value is the free space available.

The following SQL statement shows how to determine the free space that each dbspace requires:

```

DATABASE sysmaster;
SELECT partdbsnum(partnum) dbspace_num,
       trunc(count(*) * 2000) free_space_req
  FROM sysdatabases
 GROUP BY 1
 ORDER BY 1;

```

The following SQL statement queries the **syschunks** table and displays the free space available for each dbspace:

```

SELECT dbsnum dbspace_num, sum(nfree) free_space_avail
  FROM syschunks
 GROUP BY 1
 ORDER BY 1;

```



Important: *If less free space is available than the dbspace requires, either move a table from the dbspace to another dbspace or add a chunk to the dbspace.*

The dbspace estimates could be higher if you have an unusually large number of SPL routines or indexes in the database.

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of your source database server. Keep the copies available to use later. Save the configuration files that [Figure 4-1](#) lists, if they exist.

Figure 4-1
IUS, IDS 7.3, or ODS Configuration Files

UNIX	Windows NT
\$INFORMIXDIR/etc/\$ONCONFIG	%INFORMIXDIR%\etc\ONCONFIG
\$INFORMIXDIR/etc/ONCONFIG.std	%INFORMIXDIR%\etc\ONCONFIG.std
\$INFORMIXDIR/etc/sm_versions	%INFORMIXDIR%\etc\sm_versions
\$INFORMIXDIR/aaodir/adtcfg	%INFORMIXDIR%\aaodir\adtcfg.*
\$INFORMIXDIR/dbssodir/adtmasks	%INFORMIXDIR%\dbssodir\adtmasks.*
\$INFORMIXDIR/etc/sqlhosts	
\$INFORMIXDIR/etc/tctermcap	
\$INFORMIXDIR/etc/termcap	

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

UNIX:	\$INFORMIXDIR/etc/ixbar.<servernum>
Windows NT	\$INFORMIXDIR\etc\ixbar.<servernum>

If you use ON-Archive to back up your source database server and the logical logs, you also need to save copies of the configuration files in the following list:

- **\$INFORMIXDIR/etc/\$ARC_CONFIG**
- **\$INFORMIXDIR/etc/config.arc**
- **\$INFORMIXDIR/etc/oper_deflt.arc**

◆

The Windows NT version of Dynamic Server 2000 does not use ON-Archive, so you do not need to copy these files on Windows NT. ◆

UNIX

WIN NT

Stop Enterprise Replication

If your database server uses Enterprise Replication, you need to stop any applications that are doing replicatable transactions and then check that both the TRG Send and the Control Send queues are empty.

To stop Enterprise Replication

1. Stop the applications from performing replicatable transactions.
2. Verify that the TRG Send and the Control Send queues are empty.

To verify this, run the **onstat -g grp** and **onstat -g rqm** commands, as the following examples show:

```
% onstat -g grp
```

```
Informix Dynamic Server Version 7.30.UC3 -- On-Line -
- Up 00:28:15 -- 18752 Kbytes
Grouper:
Last Idle Time: 98/11/09 15:12:01
Log update buffers: 1024
Log update buffers in use: 0
```

Log update buffers in use **should be zero.**

```
% onstat -g rqm
```

```
RQM Statistics for Queue #3
Database name:      syscdr
Table name:         control_sendq
Index name:         control_sendq_key
Flags:              0x00000301
Elements in memory: 0
Elements on disk only: 0
Memory used for data: 0 Bytes
Total memory used:  0 Bytes
Element high water mark: 2000
Data high water mark: 140000 Bytes
Elements stored on disk: 0
```

Elements in memory **and** Elements stored on disk **should be zero.**

3. Stop Enterprise Replication with the following command:

```
% cdr stop
```

Close All Transactions and Shut Down the Source Database Server

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and shut down your source database server. This lets users exit and shuts down the database server gracefully. If necessary, you can perform an immediate shutdown of the database server.

Prior to upgrading the source system to the target system, make sure that no open transactions exist. Otherwise, fast recovery will fail when rolling back open transactions during the upgrade.

To let users exit and shut down the database server gracefully

1. Execute the **onmode -sy** command to put the database server in quiescent mode.
2. Wait for all users to exit.
3. Execute the **onmode -l** command to move to the next logical log.
4. Execute the **onmode -c** to force a checkpoint.
5. Make a level-0 backup of the database server.
6. Execute the **onmode -yuk** command to shut down the system.

To perform an immediate shutdown of the database server

```
onmode -l  
onmode -c  
onmode -ky
```

Check for Any Open Transactions

A shutdown procedure does not guarantee to roll back all open transactions. To guarantee that Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x has no open transactions prior to the upgrade, you need to put the source database server in quiescent mode. Execute the following command to enter quiescent mode and initiate a fast recovery:

```
oninit -s
```

UNIX

On UNIX, The **oninit -s** command rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. For more information about fast recovery, refer to your *Administrator's Guide*.

You need to execute **oninit -s** before you initialize Dynamic Server 2000. If Universal Server, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x is not left in quiescent mode, you will receive the following error when you attempt to initialize the database server, and it goes off-line:

```
Open transaction detected when changing log versions.
```



After you put the database server in quiescent mode and initiate fast recovery, issue the **onmode -yuk** command again to shut down the database server. Then review **online.log** for any possible problems and fix them.

Only after proper shutdown can you bring the new database server (Dynamic Server 2000) through the upgrade path. Any open transaction during the upgrade will cause an execution failure in fast recovery.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data before you make a level-0 backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain the database names, use the following statements with DB-Access:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

Figure 4-2 lists the commands that verify the data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 4-2
*Commands for
Verifying the Data
Integrity*

For information on **oncheck**, refer to your *Administrator's Guide*.

Verify the Database Server Mode

Before you make a backup, execute the following command to verify that your source database server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of your source database server. Figure 4-3 shows that the database server is in quiescent mode.

```
INFORMIX-OnLine Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx

OnLine Dynamic Server is in quiescent mode.
-- xxxx Kbytes
```

Figure 4-3
*Example of onstat
Status Line*

Make a Final Backup of the Source Database Server

Use ON-Bar, **ontape**, or ON-Archive to make a level-0 backup of the source database server. After you make a level-0 backup, also perform a complete backup of the logical log, including the current logical-log file. Be sure to retain and properly label the tape volume that contains the backup. For more information about making backups, refer to the *Backup and Restore Guide*.

UNIX

If you use ON-Archive, execute the following command to make a full-system, level-0 backup:

```
Onarchive> ARCHIVE/DBSPACESET=*
```



WIN NT

Do not use ON-Archive before you migrate to Windows NT. ♦

***Important:** Make a final backup for each source database server instance that you plan to upgrade.*



Verify That the Source Database Server Is Off-Line

The source database server must be off-line because the target database server uses the same files. You cannot install the target database server if any of the files that it uses are active.

Check the message log to verify that you obtain the message `shared memory not initialized...` for off-line mode.

UNIX

Modify UNIX Kernel Parameters

You might need to change some of the kernel parameters for your UNIX operating system before you install Dynamic Server 2000. To reconfigure the operating system, follow the directions in the machine notes file included on your database server distribution media and the kernel-configuration instructions for your operating system.

Install Dynamic Server 2000

On UNIX, you must be logged in as user **root** to install Dynamic Server 2000. On Windows NT, you must be a member of the **Informix-Admin** group. Set the **INFORMIXDIR** environment variable to the directory where you plan to install the database server.



Warning: If you install the target database server in the same directory where the source database server resided, the installation script overwrites the older files. If you want to preserve your source database server files, you must install the target database server in a different directory. If you install the target database server in a different directory, you need to change the value of the **INFORMIXDIR** environment variable, or the older version of the database server will start up when you reboot.

Before you overwrite the source database server, you must take the following precautions:

- If you do not have the original media for the source database server, back up the **INFORMIXDIR** directory before you install Dynamic Server 2000.
- Copy the configuration and **sqlhosts** files from the **etc** directory of **INFORMIXDIR** to another location in the file system.

To install and configure Dynamic Server 2000, follow the directions in your *Installation Guide* and in your *Administrator's Guide*. The installation script installs Dynamic Server 2000 into the **INFORMIXDIR** directory specified for user **root** on UNIX or for the **Informix-Admin** group on Windows NT.



Important: Monitor the database server message log, **online.log**, during the upgrade for any error messages. If you see an error message, resolve the error condition before you continue the upgrade procedure.

Set Environment Variables

After you install Dynamic Server 2000, verify that the following environment variables are set to the correct values:

- **INFORMIXSERVER**
- **ONCONFIG**
- **PATH**
- **INFORMIXSQLHOSTS** (if used)



Important: On UNIX, the client application looks for the **sqlhosts** file in the **etc** directory in the **INFORMIXDIR** directory. However, you can use the **INFORMIX-SQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

Customize the ONCONFIG Configuration File

You can customize your ONCONFIG configuration file and environment variables to take advantage of the new features that Dynamic Server 2000 introduced. When you initialize Dynamic Server 2000, use the same configuration that the source database server used. After you observe the performance of Dynamic Server 2000, you might want to adjust the configuration.

Informix recommends that you set the ALARMPROGRAM configuration parameter to either nothing or **no_log.sh** to prevent the generation of errors if the logical log fills during the upgrade. For more details, see [“Initialize Dynamic Server 2000” on page 4-21](#). After the upgrade, change the value of ALARMPROGRAM to **log_full.sh**.

For information on how to configure Dynamic Server 2000, refer to your *Administrator’s Guide*. For information about environment variables, refer to the *Informix Guide to SQL: Reference*. For information about how to tune the configuration parameters, refer to your *Performance Guide*.

Important: Use the same values for your target database server for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used for the source database server. Also, keep the same size for physical logs and logical logs, including the same number of logical logs, and the same *sqlhosts* file.



UNIX

Add Any Communications Support Modules

You can use a Communications Support Module (CSM) with Dynamic Server 2000. After you install the CSM components, create entries in the **concsn.cfg** file and in the options field of the **sqlhosts** file to configure the CSM. For information on how to set up the CSM, refer to your *Administrator’s Guide*.

Existing client applications do not need to be recompiled or relinked if your database server does not use CSMs. If your database server uses a CSM, client applications must relink with new Informix libraries. The client applications must also have a CSM installed and configured.

UNIX

For ON-Archive, Update Configuration Files

If you use ON-Archive for your backup and restore tool on the source database server and you will continue to use ON-Archive with Dynamic Server 2000, you might need to update the ON-Archive configuration parameters.

During the installation procedure for ON-Archive, the install script checks the `$INFORMIXDIR/etc` directory for files named **config.arc** and **oper_deflt.arc**. If the files do not exist, the install script provides them. If the files already exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc** and **Oper_deflt.ar** (initial uppercase letters). Compare your current versions (**config.arc** and **oper_deflt.arc**) with the new versions (**Config.arc** and **Oper_deflt.ar**) and determine whether new or changed configuration parameters or qualifiers exist.

Install and Configure Any DataBlade Modules

After you install Dynamic Server 2000 and before you initialize the database server, install and register any DataBlade modules, supplied by Informix or third-party vendors, that you want to add to the database server. *Registration* is the process that makes the DataBlade module code available to use in a particular database. For more information on how to use DataBlade modules, refer to the DataBlade documentation.



Important: *Test Dynamic Server 2000 completely with traditional relational data before you start to use DataBlade modules. After you successfully use DataBlade modules, you can begin to use extended data types, routines, and access methods.*



Warning: *An upgrade to Dynamic Server 2000 from Universal Server 9.14 might fail if you do not install and set up DataBlade modules after installation of Dynamic Server 2000 and before initialization.*

Initialize Dynamic Server 2000



Initialize Dynamic Server 2000 without disk initialization.

Warning: Dynamic Server 2000 writes to the logical logs with the transactions that result from creating the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, Dynamic Server 2000 halts and indicates that you must back up the logical logs. After you back up the logical logs, the database server can finish building the **sysmaster** database.

You cannot use ON-Bar or ON-Archive to back up the logical logs because the database has not been upgraded yet.

If you have ALARMPROGRAM set to “**log_full.sh**” in the ONCONFIG configuration file, errors are generated as each log file fills during the upgrade. Informix recommends that you change the value of ALARMPROGRAM to either nothing or “**no_log.sh**” so that these errors are not generated.

If your logical log does fill up during the upgrade, you need to back it up with **ontape**, the only backup tool you can use at this point. Issue the **ontape -a** command.

Execute the following command to bring Dynamic Server 2000 on-line for the first time:

```
oninit
```

As Dynamic Server 2000 comes on-line for the first time, it modifies certain disk structures. This operation should extend the initialization process by only a minute or two. In the unlikely event that your disks cannot accommodate the growth in disk structures, you will find a message in the message-log file that instructs you to run **oncheck** on a table. The **oncheck** utility will tell you that you need to rebuild an index. You should rebuild the index as instructed.

Monitor the Upgrade Complete Status

Check your message log (**online.log**) for status messages that pertain to bringing Dynamic Server 2000 on-line. The upgrade process ends when the following message is written to **online.log**:

```
Conversion completed successfully
```

This message indicates that the upgrade process completed successfully, but it does not guarantee that each individual database was upgraded successfully. The message log could contain additional information regarding the success or failure of each individual database upgrade. If a particular database upgrade fails, then you should try to connect to the database to find out the exact cause of the failure.

At the end of the upgrade of each individual database, the upgrade process runs a script to update some system catalog table entries. The message log includes messages related to this script. The success or failure of the script does not prevent the usage of a database. If the script fails for a database, however, for better performance Informix recommends that you run one of the following scripts as user **informix** while connected to the database.

- For upgrading to Dynamic Server 2000 from Universal Server 9.14:
\$INFORMIXDIR/etc/dummyupds914.sql
- For upgrading to Dynamic Server 2000, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x:
\$INFORMIXDIR/etc/dummyupds7x.sql

If you encounter a failure during the upgrade, Informix recommends that you restore from backup.

The upgrade process involves two phases: internal conversion and external conversion. Internal conversion is the process of upgrading each individual database in the system. External conversion is the process of upgrading the utilities, such as ON-Bar.

For information about any messages in the message log, refer to the *Administrator's Guide*.



Important: If the message file notes problems, solve the problems before you continue to the next step.

Update Statistics

After a successful upgrade, you need to run UPDATE STATISTICS on some of the system catalog tables in your databases, as the following lists show.

For an upgrade to Dynamic Server 2000 from a 7.3x or 7.2x database server, run UPDATE STATISTICS on the following system catalog tables in Dynamic Server 2000:

- **SYSBLOBS**
- **SYSCOLAUTH**
- **SYSCOLUMNS**
- **SYSCONSTRAINTS**
- **SYSDEFAULTS**
- **SYSDISTRIB**
- **SYSFRAGAATH**
- **SYSFRAGMENTS**
- **SYSINDICES**
- **SYSOBJSTATE**
- **SYSOPCLSTR**
- **SYSPROCAUTH**
- **SYSPROCEDURES**
- **SYSROLEAUTH**
- **SYSSYNONYMS**
- **SYSSYNTABLE**
- **SYSTABAUTH**
- **SYSTABLES**
- **SYSTRIGGERS**
- **SYSUSERS**

For an upgrade to Dynamic Server 2000 from Universal Server 9.14, run UPDATE STATISTICS on the following system catalog tables in Dynamic Server 2000:

- **SYSAGGREGATES**
- **SYSAMS**
- **SYSATTRTYPES**
- **SYSBLOBS**
- **SYSCASTS**

- SYSCOLATTRIBS
- SYSCOLAUTH
- SYSCOLUMNS
- SYSCONSTRAINTS
- SYSDEFAULTS
- SYSDISTRIB
- SYSFRAGAUTH
- SYSFRAGMENTS
- SYSINDICES
- SYSLANGAUTH
- SYSOBJSTATE
- SYSOPCLASSES
- SYSOPCLSTR
- SYSPROCAUTH
- SYSPROCEDURES
- SYSROLEAUTH
- SYSROUTINELANGS
- SYSSYNONYMS
- SYSSYNTABLE
- SYSTABAMDATA
- SYSTABAUTH
- SYSTABLES
- SYSTRACEMSGS
- SYSTRIGGERS
- SYSUSERS
- SYSXTDTYPES
- SYSXTDTYPEAUTH

Verify the Integrity of the Data

After Dynamic Server 2000 finishes upgrading the system catalog tables, open each database with DB-Access and use **oncheck** to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents, system catalog tables, data, indexes, and smart large objects, as [Figure 4-4](#) shows.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>
Check smart large objects	oncheck -cs <i>sbspace_name</i>
Check smart large object plus extents	oncheck -cS <i>sbspace_name</i>

Figure 4-4
Commands for
Verifying the Data
Integrity

Make an Initial Backup of Dynamic Server 2000

Use your Dynamic Server 2000 backup and restore tool (ON-Bar or **ontape**) to make a level-0 backup of the new database server. Do not use ON-Archive. Do not overwrite the tapes that contain the final backup of the source database server. If you use ON-Bar, refer to the *Backup and Restore Guide*. If you use **ontape**, refer to the *Archive and Backup Guide*.

Important: Do not restore the backed up logical-log files from your source database server for your target database server.



Tune Dynamic Server 2000 for Performance

When you finish the level-0 backup, the migration process is complete and users can safely use Dynamic Server 2000 to access data.

After successful migration to Dynamic Server 2000, you can tune the database server to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between the source database server and Dynamic Server 2000. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on performance topics, refer to your *Performance Guide*.

Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000

This section describes the steps for reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000. For more information, see [“Reverting to Universal Server 9.10 Through 9.13 from Universal Server 9.14” on page 5-5](#), [“Reverting to an Earlier Version of OnLine Dynamic Server” on page 11-36](#), or [“Reverting to OnLine 5.0 or 4.1” on page 12-39](#).



Important: Before you revert to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000, you need to remove all Version 9.2 objects from the databases except those created by the boot scripts in the system catalog. Do not drop the objects that the **boot90.sql** and **boot901.sql** scripts created because the reversion utility uses them. For Version 7.3x, also remove any Version 9.1x features. For Version 7.2x, also remove any Version 9.1x or Version 7.3x features.

To revert to a Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x, complete the steps that the following sections describe:

1. Review the database schema to determine whether reversion is possible.
2. Save copies of the current configuration files.
3. Stop Enterprise Replication.
4. Verify the integrity of the data.
5. Back up Dynamic Server 2000.
6. Remove Version 9.2 features.
7. Run the reversion utility (**onmode -b**).
8. Modify configuration parameters.
9. Reset environment variables.
10. Remove any Communications Support Module settings.
11. Reinstall and start the target database server.
12. Update statistics.
13. Verify the integrity of the data.
14. Back up the target database server.
15. Return the target database server to on-line mode.

Determine Whether Reversion Is Possible

The easiest reversion scenario is a Dynamic Server 2000 database that does not contain any new features. Run the reversion utility and modify the values of the configuration parameters.

Review the database schema to determine whether reversion to the earlier database server is possible. Ask the following questions:

- Does the schema file contain SQL statements that the earlier database server does not support?
- Does the database contain features that the earlier database server does not support, such as long identifiers?
- Have any new SPL routines been created in Dynamic Server 2000?

To review the database schema, execute the **dbschema** utility command. The following example displays complete information about the database **db1**:

```
dbschema -d db1 -ss
```

Database reversion occurs in two phases: a check phase and an actual reversion phase. If a database cannot be reverted, the check phase would highlight it and prevent reversion.

The following restrictions apply to reversion to a 9.14, 7.3x, or 7.2x database server from Dynamic Server 2000:

1. You cannot revert a database that was created with Dynamic Server 2000.

Drop the database before you attempt reversion.

2. You cannot revert to a 9.14, 7.3x, or 7.2x database server from a 9.2 database server that has had extensions added unless you remove the extensions.

You need to remove any new data types or routines that you created either explicitly or by registering a different version of a DataBlade module.

To be able to revert, you need to downgrade any DataBlade module back to the version that was registered prior to reversion and explicitly drop any data types and routines that were created outside of any DataBlade registration. For information on how to use DataBlade modules, see the DataBlade documentation.

3. No new routines should have been created in the upgraded databases (either implicitly or explicitly).
4. No new triggers should be defined in the upgraded databases.
5. Select triggers should not be in use.
6. User-defined statistics should not be in use.
7. No long identifiers or long usernames should be in use.

Before reversion, make sure that the R-tree indexes do not use long identifiers as indexed column names, opclass names, or opclass function names.

Also, make sure that the following disk structures do not use long identifiers:

- database tablespaces (owner and database name length)
- tablespace tablespaces (owner and tablespace name length)
- dbspaces (owner and dbspace name length) and chunks (path length)

8. No storage space should have a name more than 18 characters long.
9. No in-place ALTER TABLE statement should be pending against any table.

If a user table has an incomplete in-place ALTER operation, then you need to ensure that the in-place ALTER operation is complete by running a dummy update statement against the table. If the reversion process does not complete successfully because of in-place ALTER operations, it lists all the tables that need dummy updates. You need to perform a dummy update on each of the tables in the list before you can revert to the older database server.

If an in-place ALTER operation is incomplete against a system table, run one of the following scripts while connected to the database.

- For reversion to Universal Server 9.14 from Dynamic Server 2000:
\$INFORMIXDIR/etc/dummyupds914.sql
- For reversion to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x from Dynamic Server 2000:
\$INFORMIXDIR/etc/dummyupds7x.sql

10. No fragment expressions or constraints created on the 9.2 database server should exist in the databases.

You cannot use ALTER TABLE or ALTER INDEX statements to change fragment strategies that existed before the upgrade to Dynamic Server 2000.

The following restrictions also apply to reversion to Universal Server 9.14 from Dynamic Server 2000:

1. No new routine languages should be defined in the upgraded databases.
2. No new language authorizations must have been done in the upgraded databases.
3. No new operator classes, casts, or extended types should be defined on the 9.2 database server.

The following restrictions also apply to reversion to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x from Dynamic Server 2000:

1. No semi-detached indexes should be in the databases.
2. The databases cannot have tables whose primary access method is a user-defined access method.
3. Databases cannot have typed tables.
4. Tables cannot have any user-defined type columns.
5. Tables cannot have named row types with default values.
6. All indexes must be B-tree indexes with a total key length less than or equal to 255.
7. Tables cannot have any functional or VII indexes.
8. Databases cannot use any extensibility features, including user-defined access methods, user-defined types, aggregates, routine languages, language authorizations, trace messages, trace message classes, operator classes, errors, type authorizations, and casts.



Important: You can revert to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000 only if you remove all Version 9.2 features from the databases. If a database contains any Version 9.2 feature, the reversion will fail.

UNIX

Save Copies of the Configuration Files

Save copies of the **ONCONFIG** and **concsn.cfg** files in case you decide to upgrade to Dynamic Server 2000 again. Dynamic Server 2000 uses the **concsn.cfg** file to configure CSMs.

Stop Enterprise Replication

If your database server uses Enterprise Replication, you need to stop any applications that are doing replicatable transactions and then check that the TRG Send and Control Send queues are empty.

To stop Enterprise Replication

1. Stop the applications from performing replicatable transactions.
2. Verify that the TRG Send and Control Send queues are empty.
To verify this, run the **onstat -g grp** and **onstat -g rqm** commands, as the example on [page 4-13](#) shows.
3. Stop Enterprise Replication with the following command:

```
% cdr stop
```

Verify the Integrity of the Data

Execute the following commands to verify the integrity of the data:

```
oncheck -cI database_name
oncheck -cD database_name
oncheck -cr
oncheck -cc database_name
```

Back Up Dynamic Server 2000

Before you begin the reversion, make a complete backup of Dynamic Server 2000. If you use ON-Bar, refer to the *Backup and Restore Guide*. If you use **ontape**, refer to the *Archive and Backup Guide*.

Remove Version 9.2 Features

Before you revert, remove any Version 9.2 features that Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x does not support. For Dynamic Server 7.3x, also remove any Version 9.1x features. For OnLine Dynamic Server 7.2x, also remove any Version 9.1x and Version 7.3x features.

For a list of features that you need to remove before reversion, see [“Determine Whether Reversion Is Possible” on page 4-27.](#)

Run the Reversion Utility

Dynamic Server 2000 must be running when you execute the reversion utility. The reversion utility detects and lists any remaining features that are specific to Dynamic Server 2000. You need to remove these features before reversion can complete.

Execute the reversion utility with one of the following commands:

```
onmode -b 9.14
onmode -b 7.3
onmode -b 7.2
```

After the reversion is complete, Dynamic Server 2000 is off-line. The reversion utility drops the Dynamic Server 2000 system catalog tables and restores compatibility so that users can access the data with the earlier database server. The reversion utility does not revert changes made to the layout of the data that do not affect compatibility.

For more information about the **onmode -b** command, see [Chapter 24, “Using the onmode Utility.”](#)

Modify Configuration Parameters

Replace the Dynamic Server 2000 **ONCONFIG** file with the **ONCONFIG** file that you used before you upgraded. Alternatively, you can remove the following configuration parameters that the earlier database server does not support:

- **ALLOW_NEWLINE**
- **DD_HASHMAX**
- **DD_HASHSIZE**
- **OPT_GOAL**
- **PC_HASHSIZE**
- **PC_POOLSIZE**
- **STMT_CACHE**
- **STMT_CACHE_SIZE**
- **SYSSBSPACENAME**

You might also need to adjust the values of existing configuration parameters.

Reset Environment Variables

Reset the environment variables to values that are appropriate for the earlier database server.

UNIX

Remove Any Communications Support Module Settings

If your Dynamic Server 2000 instance used CSMs, remove the **csm** option settings from the **sqlhosts** file entries for the database server. Otherwise, the older database server will return an invalid **sqlhosts** options error. Also delete the **conscsm.cfg** file.

Reinstall and Start the Target Database Server

Install and configure Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x according to the instructions in your *Installation Guide* and your *Administrator's Guide*.

Execute the **oninit -s** command to put Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x in quiescent mode.

Update Statistics

After a successful reversion, you need to run UPDATE STATISTICS on some of the system catalog tables in your databases when the database server starts.

For reversion to a 7.2x or 7.3x database server from Dynamic Server 2000, run UPDATE STATISTICS on the following system catalog tables in Dynamic Server 7.3x or OnLine Dynamic Server 7.2x:

- SYSBLOBS
- SYSCOLAUTH
- SYSCOLUMNS
- SYSCONSTRAINTS
- SYSDEFAULTS
- SYSDISTRIB
- SYSFRAGAUTH
- SYSFRAGMENTS
- SYSINDICES
- SYSOBJSTATE
- SYSOPCLSTR
- SYSPROCAUTH
- SYSPROCEDURES
- SYSROLEAUTH
- SYSSYNONYMS
- SYSSYNTABLE

- SYSTABAUTH
- SYSTABLES
- SYSTRIGGERS
- SYSUSERS

For reversion to a 9.14 database server from Dynamic Server 2000, run UPDATE STATISTICS on the following system catalog tables in Universal Server 9.14:

- SYSAGGREGATES
- SYSAMS
- SYSATTRTYPES
- SYSBLOBS
- SYSCASTS
- SYSCOLAUTH
- SYSCOLATTRIBS
- SYSCOLUMNS
- SYSCONSTRAINTS
- SYSDEFAULTS
- SYSDISTRIB
- SYSFRAGAUTH
- SYSFRAGMENTS
- SYSINDICES
- SYSLANGAUTH
- SYSOBJSTATE
- SYSOPCLASSES
- SYSOPCLSTR
- SYSPROCAUTH
- SYSPROCEDURES
- SYSROLEAUTH
- SYSROUTINELANGS

- SYSSYNONYMS
- SYSSYNTABLE
- SYSTABAMDATA
- SYSTABAUTH
- SYSTABLES
- SYSTRACEMSGS
- SYSTRIGGERS
- SYSUSERS
- SYSXTDTYPES
- SYSXTDTYPEAUTH

Verify the Integrity of the Data

Before you allow users to access the databases, check the integrity of the data. Follow the steps that [“Verify the Integrity of the Data” on page 4-15](#) describes.

Back Up the Target Database Server

After you complete the reversion, use ON-Bar, **ontape**, or ON-Archive to make a level-0 backup of Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x. For more information about making backups, if you use ON-Bar, refer to your *Backup and Restore Guide*, or if you use **ontape** or ON-Archive, refer to your *Archive and Backup Guide*.



Important: Do not overwrite the tapes that you used to back up your source database server.

Return the Target Database Server to On-Line Mode

To bring Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x on-line, execute the **onmode -m** command. The reversion is now complete, and users can access the data.

Migrating Between Versions of Universal Server

In This Chapter	5-3
Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13	5-3
R-Tree Indexes	5-4
Tables That Contain Collection Columns	5-4
Reverting to Universal Server 9.10 Through 9.13 from Universal Server 9.14	5-5

In This Chapter

This chapter describes migration between Universal Server 9.14 and earlier versions of Universal Server (Version 9.13, Version 9.12, Version 9.11, or Version 9.10). The description covers the following topics:

- Upgrading to Universal Server 9.14 from Universal Server 9.10 through 9.13
- Reverting to Universal Server 9.10 through 9.13 from Universal Server 9.14

You can use the same procedures to upgrade to or revert from 9.1x database servers earlier than Version 9.14.

Upgrading to Universal Server 9.14 from Universal Server 9.10 Through 9.13

You can upgrade to Universal Server 9.14 from an earlier database server automatically when you first access your database. The upgrade occurs when you bring up the target database server. No specific action is required.

After you upgrade, you can use **oncheck** to verify the integrity of the reserve pages, extents, system catalog tables, data, indexes, and smart large objects. [Figure 5-1](#) shows the **oncheck** verification commands.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>
Check smart large objects	oncheck -cs <i>sbspace_name</i>
Check smart large object plus extents	oncheck -cS <i>sbspace_name</i>

Figure 5-1
*Commands for
Verifying the Data
Integrity*

R-Tree Indexes

The disk organization of R-tree indexes changed in Universal Server 9.14. If you are upgrading to Universal Server 9.11, 9.12, 9.13, or 9.14 from Universal Server 9.10, you must drop and re-create your R-tree indexes. Dropping and re-creating R-tree indexes gives you larger indexes with faster searching capabilities.

Tables That Contain Collection Columns

If a table column has a collection data type, you must unload the collection data before you migrate from Universal Server 9.10, 9.11, or 9.12 and then reload the data after the migration.

Reverting to Universal Server 9.10 Through 9.13 from Universal Server 9.14

To revert to Universal Server 9.10, 9.11, 9.12, or 9.13 from Universal Server 9.14, perform the following steps:

1. Shut down Universal Server 9.14.
2. Back up Universal Server 9.14.
3. Remove any 9.14 features that the earlier database server does not support.
4. Bring up the earlier database server.

Migrating Between Universal Server and OnLine Dynamic Server 7.2x or 7.1x

In This Chapter	6-3
Preparing for Migration	6-4
Migration Guidelines.	6-4
Changes That Universal Server Introduced	6-5
Environment Variables	6-5
Configuration Parameters.	6-5
New Database Storage Spaces	6-6
New Options Syntax in the sqlhosts File or Registry Key	6-6
Global Language Support	6-7
Reversion to OnLine Dynamic Server 7.2x from Dynamic Server 7.3x	6-8
Upgrading to Universal Server from OnLine Dynamic Server 7.2x or 7.1x	6-8
Check Available Space	6-9
Save Copies of the Current Configuration Files.	6-11
Close All Transactions in the Source Database Server.	6-12
Put OnLine Dynamic Server in Quiescent Mode	6-12
Verify the Integrity of the Data	6-13
Verify the Mode	6-13
Back Up OnLine Dynamic Server	6-14
Bring OnLine Dynamic Server Off-Line	6-14
Change UNIX Kernel Parameters	6-15
Install Universal Server	6-15
Set Environment Variables	6-16
Update the ONCONFIG Configuration File	6-16
Add Any Communications Support Modules	6-16
Update the ON-Archive Configuration Files.	6-17
Bring Universal Server On-Line	6-17

Verify the Integrity of the Data	6-18
Back Up Universal Server	6-19
Tune Universal Server for Performance	6-19
Install and Configure Any Datablade Modules	6-19
Reverting to OnLine Dynamic Server 7.2x or 7.1x from	
Universal Server	6-20
Determine Whether Reversion Is Possible.	6-21
Save Copies of the Configuration Files	6-21
Verify the Integrity of the Data	6-21
Back Up Universal Server	6-22
Remove New Universal Server Features	6-22
Uninstall Detachable Modules	6-22
Remove User-Defined Routines and User-Defined Functions	6-23
Drop Indexes over 255 Bytes	6-23
Remove Extended Data Types and Smart Large Objects	6-24
Remove Features Specific to Universal Server	6-24
Remove SPL Routines Created in Universal Server	6-25
Drop Secondary Access Methods (Indexes)	6-25
Remove Virtual Tables in External Spaces	6-25
Remove Sbspaces and External Spaces	6-25
Run the Reversion Utility	6-26
Modify Configuration Parameters	6-26
Reset Environment Variables	6-27
Remove Any Communications Support Module Settings	6-27
Install and Start OnLine Dynamic Server	6-27
Verify the Integrity of the Data	6-27
Back Up OnLine Dynamic Server.	6-27
Return OnLine Dynamic Server to On-Line Mode	6-28

In This Chapter

This chapter describes the procedure to migrate between Universal Server and OnLine Dynamic Server, Version 7.2x or Version 7.1x. This chapter covers the following topics:

- Preparing for migration
- Upgrading to Universal Server from OnLine Dynamic Server 7.2x or 7.1x
- Reverting to OnLine Dynamic Server 7.2x or 7.1x from Universal Server



Important: To migrate to Universal Server from Dynamic Server 7.3x, first revert to OnLine Dynamic Server 7.2x or 7.1x. Then follow the instructions in this chapter to upgrade to Universal Server.

You can migrate automatically between Universal Server and OnLine Dynamic Server, Version 7.10.UD1 through Version 7.2x. The migration occurs when you bring up the target database server.

To migrate to Universal Server from a database server earlier than Version 7.10, first upgrade to a 7.10 or later version of OnLine Dynamic Server and then follow the instructions in this chapter to upgrade to Universal Server.

UNIX

WIN NT

Preparing for Migration

To prepare for migration between and a 7.2x or 7.1x database server, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration.

Migration Guidelines

Informix suggests that you observe the following precautions when you migrate to Universal Server from OnLine Dynamic Server:

- Check the release notes for information about the proper operating-system release and any patches that you need for successful installation and operation of the database server.

The release notes are in one of the following directories:

- `$INFORMIXDIR/release/en_us/0333`. ♦
- `%INFORMIXDIR%\release\en_us\0333`

Release Notes appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar. ♦

On UNIX, retain both versions of the Informix product software on disk (if you have enough disk resources). On Windows NT, for a Version 7.2x or earlier database server, you cannot retain two versions of the Informix product on disk.

- Retain the installation media from both versions of the Informix product software.
- Perform a level-0 backup of all dbspaces and blobspaces with OnLine Dynamic Server. After you complete the migration, perform another level-0 backup with Universal Server.
- Use a test instance of your database server to test the installation and migration procedures.

For additional information, refer to both the *INFORMIX-Universal Server Installation Guide* and the *Getting Started with INFORMIX-Universal Server* manual.

Changes That Universal Server Introduced

This section describes the changes that Universal Server introduced that affect either migration or initial configuration.

Environment Variables

Universal Server introduced the following new environment variables that were not available in OnLine Dynamic Server, Version 7.2:

- INFORMIXCONCSMCFG
- INFORMIXKEYTAB

Review the descriptions of these environment variables for communications support services (CSM) to determine whether you need to set them. The *Informix Guide to SQL: Reference* describes these environment variables.

Configuration Parameters

Universal Server introduced new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. The *INFORMIX-Universal Server Administrator's Guide* describes the following new configuration parameters for Universal Server:

- HETERO_COMMIT
- SBSPACENAME
- VPCLASS

The SBSPACENAME parameter specifies a default *sbspace* for the storage of *smart large objects*. For more information on smart large objects and sbspaces, refer to [“New Database Storage Spaces” on page 6-6](#) and to your *Administrator's Guide*.

The VPCLASS parameter combines options from several virtual-processor configuration parameters and enables you to configure a virtual processor class with a single parameter. Consequently, if you use the VPCLASS parameter, be aware that it affects the following parameters, which you might need to change or delete:

- AFF_NPROCS
- AFF_SPROC
- MULTIPROCESSOR
- NOAGE
- NUMAIOVPS
- NUMCPUVPS
- SINGLE_CPU_VP

New Database Storage Spaces

Universal Server introduced the following two new storage spaces:

- Sbspace
- External space

An sbspace provides storage space for smart large objects. An external space is a space outside the direct control of Universal Server, and one that contains data that you want to include in a database. If you use external spaces, you must also define the access methods that Universal Server will use to retrieve the external data. For more information on external spaces, refer to your *Administrator's Guide*. For more information on user-defined access methods, refer to the *Virtual-Table Interface Programmer's Manual*.

New Options Syntax in the sqlhosts File or Registry Key

Universal Server 9.13 introduced formatting changes and new options in the **sqlhosts** file or registry key. The fifth field in the **sqlhosts** fields, the **options** field, was modified to accommodate new options and future options.

The following table shows the **sqlhosts** file or registry key fields:

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5
dbservername	nettype	hostname	servicename	options

The **dbservername** in field 1 is a key for connectivity information in the remaining fields in the **sqlhosts** file or registry key.

The **options** field contains columns. To separate columns, a comma or white space represents the end of each column. Client and database server applications check each column to determine whether the database server release supports the option.

For Universal Server, Version 9.13 or higher, the **sqlhosts** file or registry key contains a line for each connection type that the database server provides and for each instance of Universal Server to which the client application connects.



***Tip:** If you maintain more than one version of the database server, use separate **sqlhosts** files for older versions of the database server. Alternatively, you can use a separate entry with an alias for each database server.*



***Important:** Informix recommends that you use field 5, **options**, for Universal Server, Version 9.13 or Version 9.14, for the following options only: **b**, **k**, **r**, **s**. If you do not want any of these options but do want other options, use **k=1** in column 5, which is the default. Universal Server, Version 9.13 or Version 9.14, allows key values with more than one character. Place other options in subsequent columns.*

If you use multiple versions of the database server, you must maintain two separate **sqlhosts** files. For more information on the components of the **sqlhosts** file or on how to define two **sqlhosts** files, see your *Administrator's Guide*.

GLS

Global Language Support

Universal Server incorporates GLS, which enables Universal Server to handle different languages, cultural conventions, and code sets for Asian, European, Latin American, and Middle Eastern countries. The *Informix Guide to GLS Functionality* provides a full description of GLS.

Reversion to OnLine Dynamic Server 7.2x from Dynamic Server 7.3x

Before you can migrate to Universal Server from a 7.3 database server, you need to revert to a 7.2 database server. See the reversion information in [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#) After you revert to OnLine Dynamic Server, follow the instructions in this chapter for upgrading to Universal Server.

Upgrading to Universal Server from OnLine Dynamic Server 7.2x or 7.1x

This section describes the migration procedures to Universal Server from OnLine Dynamic Server, Version 7.2x or Version 7.1x.

When you migrate to Universal Server, you can install and test a database server instance with the same configuration files, environment variables, and **sqlhosts** information that you used for your source database server. After you install Universal Server and verify that it works, you might want to modify configuration files and environment variables to take advantage of the Universal Server features. For more information, refer to your *Getting Started* manual and your *Administrator's Guide*.

When you migrate to Universal Server, complete the following steps, which the sections that follow describe in more detail:

1. Check available space.
2. Save copies of the current configuration files.
3. Close all transactions in the source database server.
4. Put OnLine Dynamic Server in quiescent mode.
5. Verify the integrity of the data.
6. Verify the mode.
7. Make a final (level-0) backup.
8. Bring OnLine Dynamic Server off-line.

Important: Repeat steps 2 through 8 for each instance of OnLine Dynamic Server that you are migrating to Universal Server.



9. Change UNIX kernel parameters, if necessary.
10. Install Universal Server.
11. Verify that environment variables are set correctly.
12. Update the ONCONFIG configuration files.
13. Add any Communications Support Modules.
14. Update the backup and restore configuration parameters.
15. Bring the target database server on-line.
16. Verify the integrity of the data.
17. Make an initial (level-0) backup under your target database server.
18. Tune Universal Server for performance.

Important: Repeat steps 12 through 18 for each instance of Universal Server.

19. Install and configure any DataBlade modules that you are adding to Universal Server.



UNIX

Universal Server requires 3000 free pages of logical-log space (approximately 6000 kilobytes for a 2-kilobyte page size) to build the **sysmaster** database on UNIX. ♦

WIN NT

Universal Server requires 1500 to 3000 free pages of logical-log space (approximately 6000 kilobytes for a 4-kilobyte page size) to build the **sysmaster** database on Windows NT. ♦

Universal Server requires approximately 2000 kilobytes more space per database than OnLine Dynamic Server. The extra space is used for new system catalog tables and built-in functions that support the extensibility features of Universal Server.

When you initialize Universal Server on an existing OnLine Dynamic Server root dbspace, the database server automatically upgrades the **sysmaster** database. Each database is upgraded individually when it is first accessed. For a successful upgrade of each database, you must ensure that the extra 2000 kilobytes per database is available in each dbspace, as follows:

1. Calculate the amount of free space that each dbspace requires. In the following equation, n is the number of databases in the dbspace and X is the amount of free space they require:

$$X \text{ kilobytes free space} = 2000 \text{ kilobytes} * n$$

2. Check the amount of free space in each dbspace to determine whether you need to add more space.

Use the following SQL statements to determine the free space required and the free space available. These statements return the free-space calculation in page-size units. The **free_space_req** column value is the free space required and the **free_space_avail** column value is the free space available.

The following SQL statement shows how to determine the free space that each dbspace requires:

```
DATABASE sysmaster;
SELECT partdbsnum(partnum) dbspace_num,
       trunc(count(*) * 2000) free_space_req
  FROM sysdatabases
 GROUP BY 1
 ORDER BY 1;
```

The following SQL statement queries the **syschunks** table and displays the free space available for each dbspace:

```
SELECT dbsnum dbspace_num, sum(nfree) free_space_avail
  FROM syschunks
 GROUP BY 1
 ORDER BY 1;
```



Important: *If less free space is available than the dbspace requires, either move a table from the dbspace to another dbspace or add a chunk to it.*

The dbspace estimates could be higher if you have an unusually large number of SPL routines or indexes in the database.

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of your source database server. Keep the copies available to use later. Save the configuration files that [Figure 6-1](#) lists, if they exist.

Figure 6-1
OnLine Dynamic Server Configuration Files

UNIX	Windows NT
\$INFORMIXDIR/etc/\$ONCONFIG	%INFORMIXDIR%\etc\ONCONFIG
\$INFORMIXDIR/etc/ONCONFIG.std	%INFORMIXDIR%\etc\ONCONFIG.std
\$INFORMIXDIR/etc/sm_versions	%INFORMIXDIR%\etc\sm_versions
\$INFORMIXDIR/aaodir/adtcfg	%INFORMIXDIR%\aaodir\adtcfg.*
\$INFORMIXDIR/dbssodir/adtmasks	%INFORMIXDIR%\dbssodir\adtmasks.*
\$INFORMIXDIR/etc/sqlhosts	
\$INFORMIXDIR/etc/tctermcap	
\$INFORMIXDIR/etc/termcap	

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

UNIX: \$INFORMIXDIR/etc/ixbar.<servernum>
Windows NT: \$INFORMIXDIR\etc\ixbar.<servernum>

UNIX

If you use ON-Archive to back up and restore your source database server and the logical logs, you must also copy and save the configuration files in the following list:

- \$INFORMIXDIR/etc/\$ARC_CONFIG
- \$INFORMIXDIR/etc/config.arc
- \$INFORMIXDIR/etc/oper_deflt.arc ♦

WIN NT

The Windows NT version of Universal Server does not use ON-Archive. Therefore, you do not need to copy these files on Windows NT. ♦

Close All Transactions in the Source Database Server

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and shut down your source database server. This lets users exit and shuts down the database server gracefully. If necessary, you can perform an immediate shutdown of the database server.

To let users exit and shut down the system gracefully

1. Execute the **onmode -sy** command.
2. Wait for all users to exit.
3. Execute the **onmode -ky** command.

To perform an immediate shutdown

```
onmode -ky
```

Put OnLine Dynamic Server in Quiescent Mode

Execute the following command to enter quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

The **oninit -s** option rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. (For more information about fast recovery, refer to your *Administrator's Guide*.)

You must execute **oninit -s** before you initialize Universal Server. If the system is not left in a quiescent state, you receive the following error when you attempt to initialize the database server and it goes off-line:

```
Open transaction detected when changing log versions.
```

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain the database names, use the following statements with DB-Access:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

Figure 6-2 lists the commands that verify data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 6-2
*Commands for
Verifying the Data
Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

Verify the Mode

Before you make a backup, execute the following command to verify that your source database server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of your source database server. Figure 6-3 shows that the database server is in quiescent mode.

INFORMIX-OnLine Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx

OnLine Dynamic Server is in quiescent mode.

-- xxxx Kbytes

Figure 6-3
*Example of onstat
Status Line*

UNIX

WIN NT

Back Up OnLine Dynamic Server

Use ON-Bar, **ontape**, or ON-Archive to make a level-0 backup of OnLine Dynamic Server. After you make a level-0 backup, also perform a complete backup of the logical log, including the current logical-log file. Be sure to retain and properly label the tape volume that contains the backup. For more information about making backups, refer to your *Backup and Restore Guide*.

If you use ON-Archive, execute the following command to make a full-system, level-0 backup:

```
Onarchive> ARCHIVE/DBSPACESET=*
```

◆

Do not use ON-Archive before you migrate to Windows NT. ◆

Bring OnLine Dynamic Server Off-Line

Execute the following command to bring the source database server to off-line mode:

```
onmode -ky
```

Bring the source database server off-line to ensure that all common files are inactive. The source database server must be off-line because the target database server uses the same files. You cannot install the target database server if any of the files that it uses are active.

After you shut down the source database server, execute the following command to verify that it is off-line:

```
onstat -
```

Verify that you obtain the message `shared memory not initialized...` for off-line mode.

Important: Make a final backup for each source database server instance that you plan to upgrade.



UNIX

Change UNIX Kernel Parameters

You might need to change some of the kernel parameters for your UNIX operating system before you install Universal Server. To reconfigure the operating system, follow the directions in the machine notes file included on your database server distribution media and the kernel-configuration instructions for your operating system. For information on the location of the machine notes file, refer to “[Documentation Notes, Release Notes, and Machine Notes](#)” on page 22 of the Introduction.

Install Universal Server

On UNIX, you must be logged in as user **root**. On Windows NT, you must be a member of the **Informix-Admin** group to install Universal Server. Set the **INFORMIXDIR** environment variable to the directory where you plan to install Universal Server.



Warning: *If you install the target database server in the same directory where the source database server resided, the installation script overwrites the older files. If you want to preserve your source database server files, you must install the target database server in a different directory.*

Before you overwrite the source database server, you must take the following precautions:

- If you do not have the original media for the source database server, back up the **INFORMIXDIR** directory before you install Universal Server.
- Copy the configuration files in the **etc** directory of **INFORMIXDIR** to another location on the file system.

Follow the directions in your *Installation Guide* to install Universal Server. The installation script installs Universal Server into the **INFORMIXDIR** directory specified for user **root** on UNIX or for the **Informix-Admin** group on Windows NT. The installation script does not bring the target database server on-line.

For information on how to install Universal Server on Windows NT, refer to your *Administrator's Guide*. ♦

WIN NT

Set Environment Variables

After you install Universal Server, make sure that the following environment variables are set to the correct values:

- INFORMIXSERVER
- ONCONFIG
- PATH
- INFORMIXSQLHOSTS (if used)



Important: The client application looks for the **sqlhosts** file or registry in the **etc** directory in the **INFORMIXDIR** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

Update the ONCONFIG Configuration File

You can customize your ONCONFIG configuration file and environment variables to take advantage of the new features that Universal Server introduced. After you observe the performance of Universal Server, you might want to make further adjustments.

For information on how to configure Universal Server, refer to your *Administrator's Guide*. For information about environment variables, refer to the *Informix Guide to SQL: Reference*. For information about how to tune the configuration parameters, refer to your *Administrator's Guide*.



Important: Use the same values for your target database server for **ROOTOFFSET**, **ROOTSIZE**, and **ROOTPATH** that you used for your source database server.

UNIX

Add Any Communications Support Modules

You can use the Communications Support Module (CSM) with Universal Server. After you install the CSM components, create entries in the **concsbm.cfg** file and in the options field of the **sqlhosts** file to configure the CSM. For information on how to set up the CSM, refer to your *Administrator's Guide*.

UNIX

Existing client applications do not need to be recompiled or relinked if your database server does not use CSMs. If your database server uses a CSM, client applications must relink with new Informix libraries. The client applications must also have a CSM installed and configured.

Update the ON-Archive Configuration Files

If you use ON-Archive for your source database server backup and restore tool, and you will continue to use it with Universal Server, you might need to update ON-Archive configuration parameters.

During the installation procedure for ON-Archive, the install script checks the \$INFORMIXDIR/etc directory for files named **config.arc** and **oper_deflt.arc**. If the files do not exist, the install script provides them. If the files already exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc** and **Oper_deflt.arc** (note the initial uppercase letters). Compare your current versions (**config.arc** and **oper_deflt.arc**) with the new versions and determine whether new or changed configuration parameters or qualifiers exist.

Bring Universal Server On-Line

Execute the following command to bring Universal Server on-line for the first time:

```
oninit
```

As Universal Server comes on-line for the first time, it modifies certain disk structures. This operation should extend the initialization process by only a minute or two. In the unlikely event that your disks cannot accommodate the growth in disk structures, you will find a message in the message-log file that instructs you to run **oncheck** on a table. The **oncheck** utility will tell you that you need to rebuild an index. You should rebuild the index as instructed.



Warning: Universal Server writes to the logical log with the transactions that result from creating the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, Universal Server halts and indicates that you must back up the logical log. After you back up the logical log, Universal Server can finish building the **sysmaster** database.



Check your Universal Server message log for status messages that pertain to bringing Universal Server on-line. For information about any messages in the message log, refer to your *Administrator's Guide*.

Important: If the message file notes problems, solve the problems before you continue to the next step.

Databases under Universal Server contain almost twice as many system catalog tables than databases under OnLine Dynamic Server. The combined size of the system catalog tables has grown about 2000 kilobytes per database. You might need to account for this growth if you have many databases and limited disk space.

Verify the Integrity of the Data

After Universal Server finishes upgrading the system catalog tables, open each database with DB-Access and use **oncheck** to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents, system catalog tables, data, indexes, and smart large objects, as [Figure 6-4](#) shows.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>
Check smart large objects	oncheck -cs <i>sbspace_name</i>
Check smart large object plus extents	oncheck -cS <i>sbspace_name</i>

Figure 6-4
Commands for
Verifying the Data
Integrity

Back Up Universal Server

Use your Universal Server backup and restore tool (ON-Bar or **ontape**) to make a level-0 backup. Do not use ON-Archive. Do not overwrite the tapes you used earlier when you made your final backup of your source database server. If you use **ontape**, refer to the *INFORMIX-Universal Server Archive and Backup Guide*. If you use ON-Bar, refer to the *INFORMIX-Universal Server Backup and Restore Guide*.



Important: Do not restore the backed up logical-log files from your source database server for your target database server.

Tune Universal Server for Performance

When you finish the level-0 backup, the migration process is complete and users can use Universal Server to access data safely.

After you successfully migrate to Universal Server, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between OnLine Dynamic Server and Universal Server. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on performance topics, refer to your *Performance Guide*.

Install and Configure Any DataBlade Modules

After you successfully migrate to Universal Server, install and register any DataBlade modules, supplied by Informix or third-party vendors, that you want to add to Universal Server. *Registration* is the process that makes the DataBlade module code available to use in a particular database. For more information on how to use DataBlade modules, refer to the DataBlade documentation.



Important: Completely test Universal Server with traditional relational data before you start to use DataBlade modules. After you successfully use DataBlade modules, you can begin to use extended data types, routines, and access methods.

Reverting to OnLine Dynamic Server 7.2x or 7.1x from Universal Server

Before you can revert to OnLine Dynamic Server 7.2x or 7.1x, you need to remove all Universal Server objects from the databases except those objects that the boot scripts created in the system catalog. Do not drop any objects that the **boot90.sql** and **boot901.sql** scripts created because the reversion utility uses these objects.

To revert to Dynamic Server 7.3x from Universal Server, you need to revert to OnLine Dynamic Server 7.2x first and then upgrade the 7.2x database server to Version 7.3x. Follow the instructions for upgrading to Dynamic Server 7.3x in [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#)

This section describes the steps for reverting to OnLine Dynamic Server 7.2x or 7.1x from Universal Server. The following sections describe these steps.

1. Review the database schema to determine whether reversion is possible.
2. Save copies of the current configuration files.
3. Verify the integrity of the data.
4. Back up the Universal Server database.
5. Remove Universal Server features.
6. Run the reversion utility (**onmode -b**).
7. Modify configuration parameters.
8. Reset environment variables.
9. Remove CSM settings.
10. Start the desired version of your target database server.
11. Verify the integrity of the data.
12. Back up the target database server.
13. Return the target database server to on-line mode.

Determine Whether Reversion Is Possible

The easiest reversion scenario is a Universal Server database that does not contain any new features. Run the **onmode -b** reversion utility and modify the values of the configuration parameters.

Review the database schema to determine whether reversion to OnLine Dynamic Server 7.2x or 7.1x is possible. Does the schema file contain SQL statements that OnLine Dynamic Server does not support? Does the database contain features that OnLine Dynamic Server does not support, such as DataBlade modules, smart large objects, user-defined data types, and user-defined routines? Have any new SPL routines been created in Universal Server?

To review the database schema, execute the **dbschema** utility. For a description of the **dbschema** syntax, see [Chapter 22, “Using the dbschema Utility.”](#) The following command displays complete information about the database **db1**:

```
dbschema -d db1 -ss
```



Important: You can revert to OnLine Dynamic Server 7.2x or 7.1x from Universal Server only if you remove all Universal Server features from the databases. If a database contains any Version 9.1x feature, reversion to OnLine Dynamic Server will fail.

UNIX

Save Copies of the Configuration Files

Save copies of the **ONCONFIG** and **concsn.cfg** files in case you decide to upgrade to Universal Server again. Only Universal Server uses the **concsn.cfg** file that is used to configure CSMS.

Verify the Integrity of the Data

Execute the following commands to verify the integrity of the data:

```
oncheck -cI database_name
oncheck -cD database_name
oncheck -cr
oncheck -cc database_name
```

Back Up Universal Server

Before you begin the reversion, make a complete backup. If you use **ontape** or ON-Archive, refer to your *Archive and Backup Guide*. If you use ON-Bar, refer to your *Backup and Restore Guide*.

Remove New Universal Server Features

Before you revert, you must remove any Universal Server features that OnLine Dynamic Server does not support, as follows:

- DataBlade modules
- User-defined routines and user-defined functions
- Indexes over 255 bytes
- Extended data types
- Smart large objects
- Features specific to Universal Server
- SPL routines created under Universal Server
- Secondary access methods
- Virtual tables in extspaces
- Sbspaces and extspaces

Uninstall Detachable Modules

Use BladeManager to unregister all DataBlade modules. When you unregister DataBlade modules, you remove all data types and routines that the DataBlade modules define.

UNIX

BladeManager is a command-line utility (**blademgr**) stored in the **\$INFORMIXDIR/bin** directory. ♦

WIN NT

BladeManager is a command-line utility (**blademgr**) stored in the **%INFORMIXDIR%\bin** directory. ♦

For more information, refer to the *BladeManager User's Guide*.

However, a few hidden tables and error messages remain in each database that BladeManager connects to. These tables contain the list of DataBlade modules and DataBlade module interfaces installed in the database. The hidden table names are **sysbldregistered**, **sysbldirequired**, **sysbldipprovided**, **sysbldobjects**, and **sysbldobjdepends**. The error messages are in the **syserrors** system catalog table and have the **sqlstate** field beginning with **UGENx**.

After you unregister the DataBlade modules, execute the following SQL script to delete the hidden tables and error messages from each database:

```
DROP TABLE sysbldregistered; --All DataBlades registered
DROP TABLE sysbldirequired; --Inter-DataBlade dependencies
DROP TABLE sysbldipprovided; --DataBlade interfaces
DROP TABLE sysbldobjects; --All objects created by DataBlade
DROP TABLE sysbldobjdepends; --Dependencies between objects
DELETE FROM syserrors WHERE sqlstate LIKE 'UGEN_';
```

Remove User-Defined Routines and User-Defined Functions

Remove all user-defined routines and functions created in Universal Server because OnLine Dynamic Server does not support them. For information on user-defined routines, refer to *Extending INFORMIX-Universal Server: User-Defined Routines*.

Drop Indexes over 255 Bytes

In OnLine Dynamic Server, the maximum key length of an index is 255 bytes. In Universal Server, the maximum key length of an index is 390 bytes. If your indexes have keys longer than 255 bytes, you must drop them before you revert the database.

Remove Extended Data Types and Smart Large Objects

Universal Server provides many data types that OnLine Dynamic Server does not support. Before you revert, drop tables, columns, views, and indexes that contain the following data types:

- Built-in data types not available in OnLine Dynamic Server:
 - BOOLEAN
 - INT8
 - LVARCHAR
 - SERIAL8
- Smart large objects (CLOB and BLOB)
- User-defined data types (OPAQUE and DISTINCT)
- Collection data types (SET, MULTISSET, LIST)
- Row data types (ROW)
- Data types provided by DataBlade modules

If you do not want to drop the tables, columns, or views, you can change the data to a type that OnLine Dynamic Server supports. For example, if the database contains one new table with user-defined data types, either delete that table or change the user-defined data types to legacy data types and drop the user-defined data types from the system catalog before you revert.

For information on Universal Server data types, refer to the *Informix Guide to SQL: Reference* and *Extending INFORMIX-Universal Server: Data Types*.

Remove Features Specific to Universal Server

Before you revert, you must remove features that are specific to Universal Server from client applications and databases. For example, OnLine Dynamic Server does not support operators and casts to any data type. For information about Universal Server features, refer to the *Informix Guide to SQL: Syntax*, *Extending INFORMIX-Universal Server: Data Types*, and the *INFORMIX-ESQL/C Programmer's Manual*.

Remove SPL Routines Created in Universal Server

Before you revert, drop all SPL routines that were created in Universal Server. OnLine Dynamic Server and Universal Server support SPL routines. However, the internal structure of an SPL routine created in Universal Server is not backward compatible and does not run under OnLine Dynamic Server. For information on SPL routines, refer to the *Informix Guide to SQL: Syntax*.

Drop Secondary Access Methods (Indexes)

Drop the following secondary access methods (indexes), which OnLine Dynamic Server does not support:

- Generic B-tree indexes on user-defined and built-in data types
- R-tree indexes on spatial data such as maps and diagrams
- Functional indexes on values returned from user-defined functions
- User-defined indexes that a DataBlade module (such as Excalibur Text DataBlade) provides

For more information on indexes, refer to your *Performance Guide*.

Remove Virtual Tables in External Spaces

Drop all virtual tables stored in external spaces and remove the access methods used to access the external data. For more information on primary access methods, refer to the *Virtual-Table Interface Programmer's Manual*.

Remove Sbspaces and External Spaces

You should already have deleted columns that contain smart large objects (BLOB and CLOB) and all virtual tables. Now delete all sbspaces and extspaces. To delete these spaces, execute the following command where *spacename* is the name of the sbspaces or extspace:

```
onspaces -d spacename
```

For more information on sbspaces and external spaces, refer to your *Administrator's Guide*.

Run the Reversion Utility

Universal Server must be running when you execute the reversion utility. The reversion utility detects and lists remaining features that are specific to Universal Server that you should remove before reversion can complete.

Execute the reversion utility with the following command to revert Universal Server to OnLine Dynamic Server:

```
onmode -b 7.2
```

After the reversion is complete, Universal Server is off-line. The reversion utility drops the Universal Server system catalog tables and restores compatibility so that users can access the data with OnLine Dynamic Server. The reversion utility does not revert changes made to the layout of the data that do not affect compatibility.

After the system catalog tables are reverted, you can install OnLine Dynamic Server or OnLine Workgroup Server 7.2, 7.21, 7.22, 7.23, or 7.24.

You can also revert to another version of the database server, such as Version 7.12, Version 7.22, or Version 7.24. For more information about the **onmode -b** command, refer to [Chapter 24, “Using the onmode Utility.”](#)

Modify Configuration Parameters

Remove the following configuration parameters, which OnLine Dynamic Server does not support:

- SBSPACENAME
- VPCLASS

You might also need to adjust the values of existing configuration parameters. Alternatively, you can replace the Universal Server **ONCONFIG** file with the OnLine Dynamic Server **ONCONFIG** file that you used before you upgraded.

UNIX

Reset Environment Variables

Reset the environment variables to values that are appropriate for OnLine Dynamic Server. Also remove the following environment variables, which OnLine Dynamic Server does not support:

- INFORMIXCONCSMCFG
- INFORMIXKEYTAB

Remove Any Communications Support Module Settings

If your Universal Server instance uses CSMs, remove the **csm** option settings from the **sqlhosts** entries for the database server. Otherwise the older database server will return an invalid **sqlhosts** options error. Also delete the **concsm.cfg** file.

Install and Start OnLine Dynamic Server

Install and configure the target database server according to the instructions in your *Administrator's Guide*.

Execute the **oninit -s** command to bring your target database server to quiescent mode.

Verify the Integrity of the Data

Before you allow users to access the databases, check the integrity of the data. Follow the steps described in [“Verify the Integrity of the Data” on page 6-13](#).

Back Up OnLine Dynamic Server

After you complete the reversion, make a level-0 backup. Use either ON-Archive or the **ontape** utility to make the backup. For information about ON-Archive and **ontape**, refer to your *Backup and Restore Guide*.

Important: Do not overwrite the tapes that you used to back up your source database server.



Return OnLine Dynamic Server to On-Line Mode

To put OnLine Dynamic Server in on-line mode, execute the **onmode -m** command. The reversion is now complete, and users can access the data.

Version 8.x Database Server Migration

- Chapter 7** **Migrating Between Extended Parallel Server and Dynamic Server AD/XP 8.21**
- Chapter 8** **Migrating Between Extended Parallel Server and a 7.3 or 7.2 Database Server**
- Chapter 9** **Migrating Between Dynamic Server AD/XP 8.21.UD4 and OnLine XPS 8.11**
- Chapter 10** **Migrating Between Dynamic Server AD/XP 8.21 and OnLine Dynamic Server 7.2**

Section III



Migrating Between Extended Parallel Server and Dynamic Server AD/XP 8.21

In This Chapter	7-3
Preparing for Migration	7-3
Migration Guidelines.	7-4
Changes in Extended Parallel Server	7-5
Environment Variables	7-5
Configuration Parameters.	7-5
Upgrade and Reversion Paths for 32-Bit and 64-Bit	
Database Servers	7-6
Storage-Manager Validation and Installation	7-7
Upgrading to Extended Parallel Server from Dynamic	
Server AD/XP 8.21.UD4	7-7
Check the Available Space	7-8
Save Copies of the Current Configuration Files.	7-9
Take the 8.21.UD4 Database Server to Quiescent Mode	7-9
Verify the Integrity of the Data	7-10
Make a Final Backup of the 8.21.UD4 Database Server	7-11
Check for Open Transactions and Shut Down the 8.21.UD4	
Database Server	7-12
Install Extended Parallel Server	7-13
Copy the 8.21.UD4 Configuration Files	7-14
Verify Environment Variable Settings on Each Node	7-14
Initialize Extended Parallel Server	7-15
Make an Initial Backup of Extended Parallel Server	7-16
Tune Extended Parallel Server for Performance.	7-17

Moving Extended Parallel Server Between 32-Bit and 64-Bit Operating Systems	7-17
Reverting to Dynamic Server AD/XP 8.21.UD4 from Enterprise Decision Server	7-17
Save Copies of the Current Configuration Files	7-18
Verify the Integrity of the Data	7-19
Back Up Extended Parallel Server	7-19
Run the Reversion Script in Check Mode	7-19
Remove Features That Extended Parallel Server Introduced	7-20
Leave Extended Parallel Server in On-Line Mode	7-21
Run the Reversion Script	7-22
Reset Environment Variables	7-25
Install Dynamic Server AD/XP 8.21.UD4	7-25
Copy and Customize Configuration Files	7-25
Initialize Dynamic Server AD/XP 8.21.UD4 Without Initializing Disks	7-26
Verify the Integrity of the Data	7-26
Back Up Dynamic Server AD/XP 8.21.UD4	7-26
Return Dynamic Server AD/XP 8.21.UD4 to On-Line Mode	7-26

In This Chapter

This chapter describes the following procedures for migrating between Extended Parallel Server, Version 8.3, and Dynamic Server with AD and XP Options, Version 8.21.UD4. The description covers the following topics:

- Preparing for migration
- Upgrading to Extended Parallel Server from Dynamic Server with AD and XP Options 8.21.UD4
- Moving Extended Parallel Server between 32-bit and 64-bit operating systems
- Reverting to Dynamic Server with AD and XP Options 8.21.UD4 from Extended Parallel Server



Important: *You can upgrade directly to Extended Parallel Server only from a Version 8.21.UD4 database server. For upgrading from an earlier Version 8.x database server, you need to upgrade to Version 8.21.UD4 first. For details, see [Chapter 9, “Migrating Between Dynamic Server AD/XP 8.21.UD4 and OnLine XPS 8.11.”](#)*

Preparing for Migration

To prepare for migration between Extended Parallel Server and Dynamic Server with AD and XP Options 8.21.UD4, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration.

Migration Guidelines

Informix recommends that you observe the following guidelines when you migrate between Extended Parallel Server and Dynamic Server with AD and XP Options 8.21.UD4:

- Check the release notes and machine notes for information about the correct operating-system release and any patches that you need for successful installation and operation of the database server.

The release notes and machine notes are in the following directory:

SINFORMIXDIR/release/en_us/0333

- Retain the installation media for both versions of the Informix product software.
- Before you upgrade to Extended Parallel Server from your 8.21.UD4 database server, make sure that no open transactions exist in the source database server. Open transactions can cause problems during the fast recovery phase when upgrading to Extended Parallel Server.

For information about how to close the source database server properly prior to migration, see [“Take the 8.21.UD4 Database Server to Quiescent Mode” on page 7-9.](#)

- Before the migration procedure, perform a level-0 backup of all dbspaces of the source database server. After you complete the migration, perform another level-0 backup of the target database server.

For information about how to perform level-0 backups, see [“Make a Final Backup of the 8.21.UD4 Database Server” on page 7-11](#) and [“Make an Initial Backup of Extended Parallel Server” on page 7-16.](#)

- Use a test instance of your database server to test the migration procedure.
- Verify storage-manager validation for the target database server.

For details, see [“Storage-Manager Validation and Installation” on page 7-7.](#)

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Changes in Extended Parallel Server

This section describes the changes in Extended Parallel Server that affect either migration or initial configuration.

Environment Variables

Extended Parallel Server introduces the following new environment variables:

- **DBCENTURY**
- **NODEFDAC**
- **SOURCE_REMOTE_SHELL**
- **XFER_CONFIG**

Configuration Parameters

Extended Parallel Server introduces new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. The *Administrator's Reference* describes the following new configuration parameters:

- **DEADLOCK_DETECTION_INTERVAL**
- **DEADLOCK_REPORT_METHOD**
- **DEADLOCK_RESOLUTION**
- **DEADLOCK_VICTIM**
- **DS_TOTAL_MEMORY**
- **PAGESIZE**
- **SENDEPDS**
- **TBLSPACE_STATS**

You might also need to adjust platform-specific parameters that specify communication buffers between coservers. For more information, refer to your UNIX machine notes.

Upgrade and Reversion Paths for 32-Bit and 64-Bit Database Servers

Extended Parallel Server offers a 32-bit or 64-bit database server and supports the following upgrade paths.

Target Database Server	Source Database Server
Extended Parallel Server, 32 bit	Dynamic Server with AD and XP Options 8.21.UD4, 32 bit
Extended Parallel Server, 64 bit	Dynamic Server with AD and XP Options 8.21.UD4, 32 bit
Extended Parallel Server, 64 bit	Dynamic Server with AD and XP Options 8.21.UD4, 64 bit
Extended Parallel Server, 64 bit	Extended Parallel Server, 32 bit

Extended Parallel Server supports the following reversion paths.

Source Database Server	Target Database Server
Dynamic Server with AD and XP Options 8.21.UD4, 32 bit	Extended Parallel Server, 32 bit
Dynamic Server with AD and XP Options 8.21.UD4, 32 bit	Extended Parallel Server, 64 bit
Dynamic Server with AD and XP Options 8.21.UD4, 64 bit	Extended Parallel Server, 64 bit
Extended Parallel Server, 32 bit	Extended Parallel Server, 64 bit

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the **ism_catalog** utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Upgrading to Extended Parallel Server from Dynamic Server AD/XP 8.21.UD4

This section describes the procedures for upgrading to Extended Parallel Server from Dynamic Server with AD and XP Options, Version 8.21.UD4.

When you migrate to Extended Parallel Server, you must install a database server instance with the same configuration files, environment variables, and **sqlhosts** information that you used for your source database server. After that, you might want to modify configuration files and environment variables to take advantage of the Extended Parallel Server features. For more information, refer to your *Getting Started* manual, your *Performance Guide*, and your *Administrator's Guide*.

To migrate to Extended Parallel Server, complete the following steps, which later sections describe in detail:

1. Check the available space.
2. Save copies of the current configuration files.
3. Take the 8.21.UD4 database server to quiescent mode.
4. Verify the integrity of the data.
5. Make a final backup of the 8.21.UD4 database server.
6. Check for open transactions and shut down the 8.21.UD4 database server.



Important: Repeat steps 1 through 6 for each instance of Dynamic Server with AD and XP Options 8.21.UD4 that you are migrating to Extended Parallel Server.

7. Install Extended Parallel Server.
8. Copy the 8.21.UD4 configuration files.
9. Verify environment variable settings on each node.
10. Upgrade to Extended Parallel Server.
11. Make an initial backup of Extended Parallel Server.
12. Tune Extended Parallel Server for performance.



Important: Repeat steps 7 through 12 for each instance of Extended Parallel Server.

The sections that follow describe these steps in detail.



Important: The upgrade procedure might not be restartable. In case of a failure, you might need to restore your source database server from a backup.

Check the Available Space

Upgrading to Extended Parallel Server does not affect the data pages. A minor amount of extra space is required for new and updated tables in the **sysmaster** database and system catalogs.

To determine the minimum amount of free space required in the 8.21.UD4 database server before upgrading to Extended Parallel Server, use the following formula:

$$freespace = (MAX_SYSIND_SIZE_BYTES + (MAX_SYSIND_NROWS * 4) + 16384) \text{ bytes}$$

In the formula, `MAX_SYSIND_SIZE_BYTES` is the size of the biggest `SYSDINDEXES` table and `MAX_SYSIND_NROWS` is the number of rows in the biggest `SYSDINDEXES` table.

Save Copies of the Current Configuration Files

Save copies of the configuration files, if they are present, for each instance of Dynamic Server with AD and XP Options 8.21.UD4. Keep the copies available for later use. The configuration files for UNIX are as follows:

- `$INFORMIXDIR/etc/$ONCONFIG`
- `$INFORMIXDIR/etc/sqlhosts`
- `$INFORMIXDIR/etc/tctermcap`
- `$INFORMIXDIR/etc/termcap`
- `$INFORMIXDIR/etc/oncfg*`
- `$INFORMIXDIR/etc/xcfg*`
- `$INFORMIXDIR/etc/sm_versions`

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file from all nodes:

`$INFORMIXDIR/etc/Bixbar_*.<servernum>`

Take the 8.21.UD4 Database Server to Quiescent Mode

Before you upgrade to Extended Parallel Server, you need to make sure that no connections exist to the source database server. Communicate to client users how long you expect the database server to be off-line for the migration.

You also need to make sure that no open transactions exist in the source database server. Fast recovery would fail when rolling back open transactions during the upgrade.

To place the source database server in quiescent mode

1. Make sure that there are no user sessions on the 8.21.UD4 database server.
2. Put the database server into quiescent mode with the following command:

```
xctl onmode -sy
```
3. Move the database server to the next logical log with the following command:

```
xctl onmode -l
```
4. Force a checkpoint with the following command:

```
xctl onmode -c
```
5. Execute the following command to verify that your database server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of the database server. [Figure 7-1](#) shows that the database server is in quiescent mode.

```
Informix Extended Parallel Server  Version  x.xx.xxx  --  Quiescent  --  Up
Extended Parallel Server is in quiescent mode.
xx:xx:xx  --  xxxx Kbytes
```

Figure 7-1
*Example of onstat
Status Line*

Verify the Integrity of the Data

Use the **onutil** utility to verify the integrity of data before you make a level-0 backup of the source database server. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain the database names, use the following statements with DB-Access:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

[Figure 7-2](#) lists the **onutil** commands you can use to verify data integrity.

Figure 7-2
Commands for Verifying Data Integrity

Action	onutil Command
Check reserve pages	onutil CHECK RESERVED;
Check extents	onutil CHECK SPACE;
Check system catalog tables	onutil CHECK CATALOGS;
Check indexes for every database in the database server	onutil CHECK INDEX WITH DATA <i>database_name</i>

Informix recommends that you also run the following **onutil** command for every table in the database server before the upgrade to make sure that all data pages are in a consistent state:

```
CHECK DATA [WITH BLOBS] IN database_name.table_name
```

Important: If any of these integrity checks fails, you have to resolve the detected integrity violations before you can proceed with the 8.3 upgrade.

You might want to test and run your queries on Extended Parallel Server and make sure they produce the same results.

For more information about **onutil**, refer to the *Administrator's Reference*.

Make a Final Backup of the 8.21.UD4 Database Server

Use ON-Bar or external tables to make a level-0 backup of the 8.21.UD4 database server. After you make a level-0 backup, perform a complete backup of the logical log, including the current logical-log file.

To back up each instance of the 8.21.UD4 database server, execute the following commands:

```
xctl onmode -l      -- to advance server to the next logical log
xctl onmode -c      -- to force a checkpoint
onbar -b -L 0       -- to take a level-0 backup
```

For more information about making backups, refer to your *Backup and Restore Guide*.

Important: Make a final backup for each source database server instance that you plan to upgrade.

Check for Open Transactions and Shut Down the 8.21.UD4 Database Server

You should have no open transactions when you upgrade to Extended Parallel Server from the 8.21.UD4 database server. To verify that no open transactions remain, run the following commands:

```
xctl onstat -g xtm  
grep IDLE
```

These commands should return the following status:

```
Coordinator state = IDLE
```

If the coordinator state is IDLE, you can shut down the database server. If the state is not IDLE, take the following steps:

1. Make sure that the 8.21.UD4 database server is in quiescent mode, as [“Take the 8.21.UD4 Database Server to Quiescent Mode” on page 7-9](#) describes.
2. Run **xctl onmode -l**.
3. Run **xctl onmode -c**.

If coordinator state is IDLE, you can shut down the database server. If coordinator state is still not IDLE, contact Informix Technical Support.

You can let users exit and shut down the database server gracefully or, if necessary, you can perform an immediate shutdown of the database server.

To let users exit and shut down the system gracefully

1. Execute the **xctl onmode -sy** command.
2. Wait for all users to exit.
3. Execute the **xctl onmode -l** command.
4. Execute the **xctl onmode -c** command.
5. Execute the **xctl onmode -yuk** command.

To perform an immediate shutdown

```
xctl onmode -ky
```

Verify that you get the message `shared memory not initialized...` for off-line mode, as follows:

```
onstat -
```

Install Extended Parallel Server

To install Extended Parallel Server, follow the instructions in your *Installation Guide*.



Warning: If you install Extended Parallel Server in the same directory where the 8.21.UD4 database server resides, the installation script overwrites the older files. If you want to preserve your 8.21.UD4 files, copy them to a different directory and then install Extended Parallel Server in the same directory.

Check the machine notes and release notes for information about the correct operating-system patches, recommended shared-memory parameters, and configurations that successful installation and operation of the database server requires. The release notes are in the following directory:

\$INFORMIXDIR/release/en_us/0333

Make any changes that the machine notes or release notes recommend on every node. For more information on how to install Extended Parallel Server, refer to your *Installation Guide* and your *Performance Guide*.

Upgrading to Extended Parallel Server from Dynamic Server with AD and XP Options 8.21.UD4 begins when you initialize the 8.3 database server. It detects an older version stamp and verifies that the last log record is an 8.21.UD4 checkpoint log record. The 8.21.UD4 database server needs to be shut down before the upgrade because any open transactions would cause the upgrade to fail in fast recovery.

Copy the 8.21.UD4 Configuration Files

Before you start the upgrade to Extended Parallel Server, copy the following files to the 8.3 installation from the 8.21.UD4 installation:

- **\$INFORMIXDIR/etc/oncfg***
- **\$INFORMIXDIR/etc/xcfg***
- **\$INFORMIXDIR/etc/SONCONFIG**
- **\$INFORMIXDIR/etc/sqlhosts**

Informix recommends that the 8.3 **ONCONFIG** file be identical to the 8.21.UD4 **ONCONFIG** file before you start the upgrade to Extended Parallel Server, except for necessary changes. You cannot change the values for some of the **ONCONFIG** parameters until after the upgrade is complete.

After the upgrade, you can customize your configuration files and environment variables to take advantage of the new features that Extended Parallel Server introduced. After you observe the performance of Extended Parallel Server, you might want to make further adjustments.

For configuration information, refer to your *Administrator's Guide*. For information about environment variables, refer to the *Informix Guide to SQL: Reference*. For performance monitoring and tuning information, refer to your *Performance Guide*.

Verify Environment Variable Settings on Each Node

Verify that the following environment variables are set correctly on every node:

- **INFORMIXDIR**
- **INFORMIXSERVER**
- **ONCONFIG**
- **PATH**
- **INFORMIXSQLHOSTS**

The client applications look for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

Set the environment variable **PATH** so that the local directory is searched before the **INFORMIXDIR** directory. The installation script installs Extended Parallel Server into the **INFORMIXDIR** directory specified for user **root** or **informix**.

You can change the values of these environment variables after you install Extended Parallel Server and before you invoke DB-Access.



Important: *If continuous logs backup was configured on the 8.21.UD4 database server, you either have to switch logs backup to off by setting it to NONE in the **\$INFORMIXDIR/etc/\$ONCONFIG** file or have a validated storage manager installed and configured for Extended Parallel Server. Failure to take one of these actions might cause upgrade errors.*

Initialize Extended Parallel Server

Using the same configuration parameters as for the 8.21.UD4 database server, initialize Extended Parallel Server without initializing the disks. To do this, issue the following command:

```
xctl -C oninit
```



Warning: *Do not use **oninit -iy** because the **-iy** option would initialize your disks, which would erase your data.*

Log messages that refer to the upgrade appear in the message log file. The message log should contain transaction log records. The results of the upgrade for each subsystem appear in the file until the upgrade is complete.

To view the message log, run the following command:

```
onstat -m
```

The following message indicates the end of the upgrade process:

```
Conversion to 8.3 completed successfully.
```

If the **sysmaster** database takes a while to build, the log file contains the following message:

```
[001 11:48:33 External conversion thread is waiting for
'sysmaster' database to be converted/built.
This thread will wait for approximately 10
minutes before aborting.
```

After the upgrade is complete, review the message log to verify that your **sysmaster** and **sysutils** databases were created successfully.

Also check the message log for information about coservers. To verify that all your coservers are up and running, execute the following command:

```
xctl onstat -
```

If the message log reports any upgrade failures, contact Informix Technical Support.

For more information about messages in the message log, refer to Appendix B, “[Upgrade and Reversion Messages](#),” and your *Administrator’s Guide*.



Important: *If the message log indicates any problems, solve each problem before you continue with the next step.*

Make an Initial Backup of Extended Parallel Server

Use the backup and restore tool (ON-Bar) to make a level-0 backup of Extended Parallel Server. The 8.21 backups are not readable by Extended Parallel Server. If the 8.3 upgrade completed successfully, you need to do a level-0 backup of the new database server instance.

Make sure that the storage manager is installed and properly configured, as “[Storage-Manager Validation and Installation](#)” on page 7-7 describes. Also make sure that the 8.21 backup tapes will not be overwritten.

You can then take the database server to quiescent mode and run the following command:

```
onbar -b -L 0
```

For more information about ON-Bar, see the *Backup and Restore Guide*.

Extended Parallel Server supports unloading directly to a pipe or to a tape drive. You do not need to unload data first to disk.

Instead of ON-Bar, you can use external tables to perform a backup. For information about loading data to and unloading data from external tables, see the *Administrator’s Reference*.



Important: *Do not restore the backed up logical-log files from Dynamic Server with AD and XP Options 8.21.UD4 for Extended Parallel Server.*

Tune Extended Parallel Server for Performance

When you finish the level-0 backup, the migration process is complete and users can use Extended Parallel Server to access data safely. After you have your applications running normally on Extended Parallel Server, consider using new 8.3 features to improve application performance.

If you created sample queries for comparison, you can use them to characterize the performance differences between Dynamic Server with AD and XP Options 8.21.UD4 and Extended Parallel Server. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on performance topics, refer to your *Performance Guide*.

Moving Extended Parallel Server Between 32-Bit and 64-Bit Operating Systems

To move Extended Parallel Server to a 64-bit operating system from a 32-bit operating system, shut down the database server cleanly and bring it back up on the 64-bit operating system. Similarly, to move Extended Parallel Server to a 32-bit operating system from a 64-bit operating system, shut down the database server cleanly and bring it back up on the 32-bit operating system.

Reverting to Dynamic Server AD/XP 8.21.UD4 from Extended Parallel Server

You can revert to Dynamic Server with AD and XP Options 8.21.UD4 from Extended Parallel Server if you upgraded to Extended Parallel Server, Version 8.3, from a Version 8.21.UD4 database server. Before the reversion, you need to remove all Version 8.3 objects from the databases except those created by the boot scripts in the system catalog.

This section describes the steps for reverting to Dynamic Server with AD and XP Options 8.21.UD4 from Extended Parallel Server:

1. Save copies of the current configuration files.
2. Verify the integrity of the data.
3. Back up Extended Parallel Server.
4. Run the reversion script in check mode.
5. Remove features that Extended Parallel Server introduced.
6. Leave Extended Parallel Server in on-line mode.
7. Run the reversion script.
8. Reset environment variables.
9. Install Dynamic Server with AD and XP Options 8.21.UD4.
10. Copy and customize configuration files.
11. Initialize Dynamic Server with AD and XP Options 8.21.UD4 without initializing disks.
12. Verify the integrity of the data.
13. Back up Dynamic Server with AD and XP Options 8.21.UD4.
14. Return Dynamic Server with AD and XP Options 8.21.UD4 to on-line mode.

The following sections describe these steps in detail.



Important: *The reversion procedure might not be restartable. In case of a failure, you might need to restore your database server from a backup.*

Save Copies of the Current Configuration Files

Save copies of the current configuration files for when you upgrade to Extended Parallel Server again.

Verify the Integrity of the Data

Execute the following commands to check the integrity of the data:

```
onutil CHECK RESERVED DISPLAY DATA
onutil CHECK SPACE DISPLAY DATA
onutil CHECK CATALOGS
onutil CHECK TABLE DATA DISPLAY DATA
onutil CHECK INDEX DISPLAY DATA
```

For details about using these commands, see [“Verify the Integrity of the Data” on page 7-10](#).

Back Up Extended Parallel Server

Before you begin the reversion, make a complete backup of Extended Parallel Server. You can use ON-Bar or external tables to make the backup. For information about using ON-Bar, see your *Backup and Restore Guide*. For information about unloading data to external tables, see the *Administrator's Reference*.

Run the Reversion Script in Check Mode

You can run the **revert_to** reversion script in check mode to list the Extended Parallel Server features that you need to remove before reverting to Dynamic Server with AD and XP Options 8.21.UD4. This script is in the **\$INFORMIXDIR/bin** directory.

To execute this script, you must be user **informix** and you must be connected to the primary coserver. Before you run the script, place Extended Parallel Server in on-line mode. While the database server is in on-line mode, do not allow any user activity because modified data would cause the check script to fail. To run the script in check mode, issue either of the following commands, depending on whether you are reverting to a 32-bit or 64-bit database server:

```
revert_to 8.21 32b check
revert_to 8.21 64b check
```

Look in the message log file for messages about the progress of the operation. The ONCONFIG parameter MSGPATH specifies the name of the message log file.

Remove Features That Extended Parallel Server Introduced

Before you revert, remove any Extended Parallel Server features that the 8.21.UD4 database server does not support, according to the instructions in the message log file. You need to remove any of the following Extended Parallel Server features:

- **Violations table**
To remove this feature, stop violations.
- **ON-Bar enhancements**
Remove ON-Bar features new in Extended Parallel Server.
- **External BLOB columns**
To remove this feature, use ALTER TABLE...DROP COLUMN statements.
- **In-place ALTER operations**
To remove this feature, perform a dummy update.
- **Triggers**
To remove triggers, use the DROP TRIGGER statement.
- **SPL routine enhancements**
- **Coarse grain locking**
- **Range clustered indexes**
To remove this feature, use the ALTER INDEX statement.
- **Range fragmentation**

You can run the reversion script in check mode to determine what features you need to remove from Extended Parallel Server. If you start the reversion without removing an 8.3 feature that is incompatible with Dynamic Server with AD and XP Options 8.21.UD4, the message log will contain messages like the following ones:

```
[001] 16:49:00 Trigger reversion test start.  
[001] 16:49:04 ** WARNING ** Please drop all triggers first and try again.  
[001] 16:49:00 Trigger reversion test failed.
```

The following message indicates that the reversion process has been cancelled due to a failure in the test phase:

```
[001] 12:23:27 Reversion Cancelled due to Reversion Test Failure.
```

Such a failure means that you need to remove one or more incompatible features before reversion can complete successfully. At this point Extended Parallel Server is in on-line mode and none of the actual reversion changes has taken place. You can remove the incompatible feature or features and attempt reversion again.

Leave Extended Parallel Server in On-Line Mode

Before you start the reversion, execute the following command to verify that Extended Parallel Server is in on-line mode:

```
onstat -
```

The first line of the **onstat** output contains the status of the database server. [Figure 7-3](#) shows that the database server is in on-line mode.

```
Informix Enterprise Decision Server Version x.xx.xxx -- On-line -- Up
Extended Parallel Server is in on-line mode.
xx:xx:xx -- xxxx Kbytes
```

Figure 7-3
*Verification of the
Database Server
Mode*

Run the Reversion Script

The reversion script, **revert_to**, performs the reversion to Dynamic Server with AD and XP Options 8.21.UD4 from Extended Parallel Server. This script is in the **\$INFORMIXDIR/bin** directory.

To execute this script, you must be user **informix** and you must be connected to the primary coserver. Before you run the script, place Extended Parallel Server in on-line mode. While the database server is in on-line mode, do not allow any user activity because modified data would cause the check script to fail. To run the script, issue either of the following commands, depending on whether you are reverting to a 32-bit or 64-bit database server:

```
revert_to 8.21 32b
```

```
revert_to 8.21 64b
```

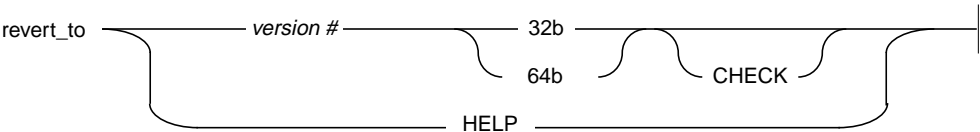
Before the reversion, the database server implicitly checks for any features that are incompatible with Version 8.21.UD4 and reports them (similar to the check mode, which [“Run the Reversion Script in Check Mode” on page 7-19](#) describes). The database server will not proceed with the actual reversion until all incompatibilities are resolved.

Messages like the following ones can appear in the message log file during reversion:

```
[001] 11:34:30 On-Line mode.
[001] 11:35:19 Reversion test started...
[001] 11:35:19 Doing internal reversion test...
[001] 11:35:24 Internal reversion test complete.
[001] 11:35:25 CM reversion test start.
[001] 11:35:26 CM reversion test completed successfully.
[001] 11:35:27 Range fragmentation reversion test start.
[001] 11:35:33 Range fragmentation reversion test completed successfully.
[001] 11:35:35 External table blob reversion test start.
[001] 11:35:40 External table blob reversion test completed successfully.
[001] 11:35:42 Range clustered indexes reversion test start.
[001] 11:35:47 Range clustered indexes reversion test completed successfully.
[001] 11:35:49 Trigger reversion test start.
[001] 11:35:54 Trigger reversion test completed successfully.
[001] 11:35:56 Violations table reversion test start.
[001] 11:36:03 Violations table reversion test completed successfully.
[001] 11:36:04 SPL routine reversion test start.
[001] 11:36:08 SPL routine reversion test completed successfully.
[001] 11:36:50 Reversion test done.
[001] 11:36:57 Local table duplication reversion start.
[001] 11:37:02 Local table duplication reversion completed successfully.
[001] 11:37:02 External reversion done. Continuing with internal reversion....
[001] 11:37:02 Reverting system catalogs for database sysmaster started.
[001] 11:37:02 Quiescent mode.
[001] 11:37:04 Reverting system catalogs for database sysmaster succeeded.
[001] 11:37:04 Reverting system catalogs for database sysutils started.
[001] 11:37:04 Reverting system catalogs for database sysutils succeeded.
[001] 11:37:04 Reverting partition header pages started.
[001] 11:37:05 On-Line mode.
[001] 11:37:06 Reverting partition header pages started succeeded.
[001] 11:37:06 Reverting tables which underwent In-Place Alter.
[001] 11:37:11 Checkpoint completed: duration was 4 seconds.
[001] 11:37:11 Starting CM safewrite reversion.
[001] 11:37:11 CM safewrite reversion successfully completed.
[001] 11:37:11 Checkpoint completed: duration was 6 seconds.
[001] 11:37:11 Reversion complete.
WARNING: A restart of the current XPS version
will undo this reversion. Install the Informix
server (8.21) you reverted to.
[001] 11:37:11 Bringing coserver down.
[001] 11:37:16 Informix Extended Parallel Server stopped.
```

When reversion is complete, the database server writes a checkpoint log record that is compatible with the Version 8.21.UD4 database server and shuts down Extended Parallel Server.

The syntax of the reversion script follows.



Element	Purpose	Key Considerations
<i>version #</i>	Specifies the version of the target database server for reversion.	For reversion from Extended Parallel Server, this value must be 8.21.
32b	Indicates that the reversion is to a 32-bit database server.	
64b	Indicates that the reversion is to a 64-bit database server.	
CHECK	Displays the source database server features that you need to remove before reversion, with instructions for removing the features.	
HELP	Displays information about the reversion script.	

Reset Environment Variables

Reset the environment variables to values that are appropriate for the 8.21.UD4 database server.

Install Dynamic Server AD/XP 8.21.UD4

Install Dynamic Server with AD and XP Options 8.21.UD4, according to the instructions in your *Installation Guide*.

Copy and Customize Configuration Files

After reversion is complete and before you bring up Dynamic Server with AD and XP Options 8.21.UD4, copy the following files to the 8.21.UD4 installation from the 8.3 installation:

- `$INFORMIXDIR/etc/oncfg*`
- `$INFORMIXDIR/etc/xcfg*`

Remove any configuration parameters that the Version 8.21.UD4 database server does not support, including the following parameters:

- `PAGESIZE`
- `TBLSPACE_STATS`
- `DS_TOTAL_MEMORY`
- `DEADLOCK_DETECTION_INTERVAL`
- `DEADLOCK_REPORT_METHOD`
- `DEADLOCK_RESOLUTION`
- `DEADLOCK_VICTIM`
- `SENDEPDS`

You might also need to adjust the values of existing configuration parameters. Alternatively, you can replace the 8.3 `ONCONFIG` file with the 8.21.UD4 `ONCONFIG` file that you used before you upgraded.

You can customize configuration files for Dynamic Server with AD and XP Options 8.21.UD4. Configure the database server according to the instructions in your *Administrator's Guide*.

Initialize Dynamic Server AD/XP 8.21.UD4 Without Initializing Disks

Using the same configuration parameters as for Extended Parallel Server, initialize Dynamic Server with AD and XP Options 8.21.UD4 without initializing the disks. To do this, issue the following command:

```
xctl -C oninit
```

Warning: Do not use **oninit -iy** because the **-iy** option will initialize your disks, which will erase your data.



Verify the Integrity of the Data

Before you allow users to access the databases, verify the integrity of the data and fix any problems. Follow the steps under [“Verify the Integrity of the Data” on page 7-10.](#)

Back Up Dynamic Server AD/XP 8.21.UD4

After you complete the reversion, make a level-0 backup. You can use ON-Bar or external tables to make the backup. For information about using ON-Bar, refer to your *Backup and Restore Guide*. For information about unloading data to external tables, see the *Administrator's Reference*.

Important: Do not overwrite the tapes that you used to back up Extended Parallel Server.



Return Dynamic Server AD/XP 8.21.UD4 to On-Line Mode

To bring Dynamic Server with AD and XP Options 8.21.UD4 on-line, execute the **xctl onmode -m** command. The reversion is now complete, and users can access the data.

Migrating Between Extended Parallel Server and a 7.3 or 7.2 Database Server

In This Chapter	8-3
Preparing for Migration	8-3
Migration Guidelines.	8-4
Changes That Extended Parallel Server Introduces	8-4
Environment Variables	8-5
Configuration Parameters.	8-5
Data Movement Utility.	8-5
Backup Tools and ON-Bar Configuration	8-6
Upgrade to OnLine Dynamic Server 7.2x from OnLine	
Dynamic Server 4.1 to 7.1x	8-6
Storage-Manager Validation and Installation	8-6
Upgrading to Extended Parallel Server from Dynamic	
Server 7.3x or OnLine Dynamic Server 7.2x.	8-7
Configure and Check Available Space	8-8
Save Copies of the Current Configuration Files.	8-9
Close All Transactions in the Source Database Server.	8-10
Put the Source Database Server in Quiescent Mode	8-11
Verify the Integrity of the Data	8-11
Verify the Mode	8-12
Back Up the Source Database Server	8-12
Change UNIX Kernel Parameters	8-13
Edit the Schema File to Suit Extended Parallel Server.	8-13
Set Environment Variables	8-13
Install Extended Parallel Server	8-14
Copy Database Server Utilities to Each Node	8-15
Update the ONCONFIG Configuration File	8-15
Bring Extended Parallel Server On-Line	8-16
Use onutil to Create Cogroups and Dbslices	8-16

Edit the onxfer Configuration File	8-17
Unload the 7.3x or 7.2x Database Schema and Data	8-17
Load the Database Schema and Data	8-17
Make an Initial Backup of Extended Parallel Server	8-17
Tune Extended Parallel Server for Performance	8-18
Reverting to Dynamic Server 7.3x or OnLine Dynamic	
Server 7.2x from Extended Parallel Server	8-18
Unloading Data to External Tables	8-19
Loading Data from External Tables	8-19

In This Chapter

This chapter describes the procedures to migrate between Extended Parallel Server, Version 8.3, and Dynamic Server, Version 7.3x, or OnLine Dynamic Server, Version 7.2x. The description covers the following topics:

- Preparing for migration
- Upgrading to Extended Parallel Server from Dynamic Server 7.3x or OnLine Dynamic Server 7.2x.
- Reverting to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x from Extended Parallel Server
- Reverting to an earlier 7.x database server from OnLine Dynamic Server 7.2x



***Tip:** To migrate to Extended Parallel Server from a 7.0x or 7.1x database server, you need to upgrade to OnLine Dynamic Server 7.2x first.*

Preparing for Migration

To prepare for migration between Extended Parallel Server and Dynamic Server 7.3x or OnLine Dynamic Server 7.2x, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration. To migrate from a 7.0x or 7.1x database server, you need to upgrade to a 7.2x database server first. The new data movement utility, **onxfer**, facilitates database schema and data transfer to Extended Parallel Server from a 7.3x or 7.2x database server.

Migration Guidelines

Informix suggests that you observe the following precautions when you migrate between Dynamic Server 7.3x or OnLine Dynamic Server 7.2x and Extended Parallel Server:

- Check the release notes and machine notes for information about the proper operating-system release and any patches that you need for successful installation and operation of the database server.

The release notes and machine notes are in the following directory:

`$INFORMIXDIR/release/en_us/0333`.

Retain both versions of the Informix product software on disk (if you have enough disk resources).

- Retain the installation media from both versions of the Informix product software.
- Perform a level-0 backup with the 7.3x or 7.2x database server. After you complete the migration, perform another level-0 backup with Extended Parallel Server.
- Use a test instance of your database server to test the installation and migration procedures.
- Verify storage-manager validation for the target database server.

For details, see, [“Storage-Manager Validation and Installation” on page 8-6](#).

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Changes That Extended Parallel Server Introduces

This section describes the changes that Extended Parallel Server introduces that affect either migration or initial configuration.

The changes that Dynamic Server with AD and XP Options, Version 8.21, introduced also affect migration to Extended Parallel Server from a 7.3x or 7.2x database server. For information about Version 8.21 changes, see [“Changes That Dynamic Server AD/XP 8.21 Introduced” on page 10-5](#).

Environment Variables

Extended Parallel Server introduces **XFER_CONFIG** and **SOURCE_REMOTE_SHELL**, new environment variables that were not available in Dynamic Server, Version 7.3x, or OnLine Dynamic Server, Version 7.2x.

The *Informix Guide to SQL: Reference* describes these environment variables. For performance implications and guidelines, refer to your *Performance Guide*.

Configuration Parameters

Extended Parallel Server introduces new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. Your *Administrator's Guide* describes the following new configuration parameters:

- MAX_CHUNKS
- MAX_DBSLICES
- MAX_DBSPACES
- PAGESIZE
- TBLSPACE_STATS

Data Movement Utility

The **onxfer** utility facilitates data movement to Extended Parallel Server from a 7.3x or 7.2x database server. This chapter includes information about using **onxfer** for unloading the 7.3x or 7.2x data and database schema and loading them into Extended Parallel Server.

This utility can unload and load data between database servers in iterative mode, when the source and target database servers are running simultaneously, or in staged mode, when the database servers are not running simultaneously. For the iterative mode, you can configure **onxfer** to unload and load a subset of the database tables.

For details about **onxfer** syntax and configuration parameters, see [Chapter 26, “Using the onxfer Utility.”](#)

Backup Tools and ON-Bar Configuration

Extended Parallel Server does not support the **ontape** utility or ON-Archive. To back up Extended Parallel Server, you need to use ON-Bar or data migration tools.

If you use ON-Bar when you migrate to Extended Parallel Server from Dynamic Server 7.3x or OnLine Dynamic Server 7.2x, you need to make substantial changes in the ON-Bar configuration. For details about how to configure ON-Bar, refer to your *Backup and Restore Guide*.

Upgrade to OnLine Dynamic Server 7.2x from OnLine Dynamic Server 4.1 to 7.1x

Before you upgrade to Extended Parallel Server from a 7.x database server earlier than OnLine Dynamic Server, Version 7.20, you need to upgrade to Version 7.2x. For instructions on how to upgrade from an earlier 7.x database server, see [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#)

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the **ism_catalog** utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Upgrading to Extended Parallel Server from Dynamic Server 7.3x or OnLine Dynamic Server 7.2x

This section describes the procedures for migrating to Extended Parallel Server from Dynamic Server, Version 7.3x, or OnLine Dynamic Server, Version 7.2x.

When you migrate to Extended Parallel Server, complete the following steps, which the sections that follow describe in more detail:

1. Configure and check available space.
2. Save copies of the current configuration files.
3. Close all transactions in the source database server.
4. Put the source database server in quiescent mode.
5. Verify the integrity of the data.
6. Verify the mode.
7. Back up the source database server.
8. Change UNIX kernel parameters.
9. Edit the schema file to suit Extended Parallel Server.
10. Set environment variables.
11. Install Extended Parallel Server.
12. Copy database server utilities to each node.
13. Update the `ONCONFIG` configuration file.
14. Bring Extended Parallel Server on-line.
15. Use **onutil** to create cogroups and dbslices.
16. Edit the **onxfer** configuration file.
17. Unload the 7.3x or 7.2x database schema and data.
18. Load the database schema and data.
19. Make an initial backup of Extended Parallel Server.
20. Tune Extended Parallel Server for performance.

Important: Repeat steps 12 through 20 for each instance of Extended Parallel Server.



Configure and Check Available Space

Configure your computer memory equally among all the nodes. Because only one **ONCONFIG** file exists in Extended Parallel Server and the configuration parameter is configured globally, the amount of memory you configure in your **ONCONFIG** file is limited to the node with the smallest amount of memory. If your system memory is configured unequally, Extended Parallel Server cannot take advantage of the nodes that have extra memory.

If you want multiple coservers in Extended Parallel Server, add a **COSERVER** section to the **ONCONFIG** file. You can copy the **onconfig.xps** template file and customize it for your database server. For more information about multiple coserver configuration, see your *Administrator's Guide* and the *Administrator's Reference*.

When you calculate disk-space requirements for Extended Parallel Server, take into account the size of your tables. You need additional space for control information for your tables (approximately 60 bytes per page). Also consider disk space for the following needs:

- Root dbspace for all your nodes
- Physical logs
- Logical-log space
(If you require logging during loads, add the additional space.)
- Temporary dbspace
- Indexes
- Summary tables and index from your on-line transaction processing (OLAP) tools
- Mirroring
- Future growth
- File-system space for your operating system and Informix products
- Staging space for loading and unloading files
- Safe write area
- New system catalog tables
- New **sysmaster** tables

For information on disk-space requirements for Extended Parallel Server, refer to your *Administrator's Guide*.

To ensure that sufficient space is available, check the amount of free space in each dbspace to determine whether you need to add more space. Use the following SQL statements to determine the free space you require and the free space available. These statements return the free-space calculation in page-size units. The **free_space_req** column value is the free space you require and the **free_space_avail** column value is the free space available.

The following SQL statement shows how to determine the free space that each dbspace requires:

```
DATABASE sysmaster;  
SELECT partdbsnum(partnum) dbspace_num,  
       trunc(count(*) * x) free_space_req  
   FROM sysdatabases  
  GROUP BY 1  
 ORDER BY 1;
```

The following SQL statement queries the **syschunks** table and displays the free space available for each dbspace:

```
SELECT dbsnum dbspace_num, sum(nfree) free_space_avail  
   FROM syschunks  
  GROUP BY 1  
 ORDER BY 1;
```



Important: *If a dbspace has less free space available than it requires, either move a table from the dbspace to another dbspace or add a chunk to it. The dbspace estimates could be higher if you have an unusually large number of SPL routines or indexes in the database.*

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of the 7.3x or 7.2x database server. Keep the copies available to use later. Save the configuration files that the following list shows, if they exist:

- **\$INFORMIXDIR/etc/\$ONCONFIG**
- **\$INFORMIXDIR/etc/ONCONFIG.std**
- **\$INFORMIXDIR/aaodir/adtcfg**
- **\$INFORMIXDIR/dbssodir/adtmasks**

- **\$INFORMIXDIR/etc/sqlhosts**
- **\$INFORMIXDIR/etc/tctermcap**
- **\$INFORMIXDIR/etc/termcap**
- **\$INFORMIXDIR/etc/\$ONCONFIG**

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

\$INFORMIXDIR/etc/ixbar.<servernum>

If you use ON-Archive to back up Dynamic Server 7.3x or OnLine Dynamic Server 7.2x and logical logs, you must also copy and save the configuration files in the following list:

- **\$INFORMIXDIR/etc/\$ARC_CONFIG**
- **\$INFORMIXDIR/etc/config.arc**
- **\$INFORMIXDIR/etc/oper_deflt.arc**

Close All Transactions in the Source Database Server

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and shut down your source database server.

To let users exit and shut down the system gracefully

1. Execute the **onmode -sy** command.
2. Wait for all users to exit.
3. Execute the **onmode -ky** command.

Put the Source Database Server in Quiescent Mode

Execute the following command to put the source database server in quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

The **oninit -s** option rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. (For more information about fast recovery, refer to your *Administrator's Guide*.)

Verify the Integrity of the Data

Use the **onutil** utility to verify the integrity of data before you make a level-0 backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain the database names, use the following statements with DB-Access:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

Figure 8-1 lists the **onutil** commands you can use to verify data integrity.

Figure 8-1
Commands for Verifying Data Integrity

Action	onutil Command
Check reserve pages	onutil CHECK RESERVED DISPLAY DATA
Check extents	onutil CHECK SPACE DISPLAY DATA
Check system catalog tables	onutil CHECK CATALOGS
Check data	onutil CHECK TABLE DATA DISPLAY DATA
Check indexes	onutil CHECK INDEX DISPLAY DATA

For information about **onutil**, refer to the *Administrator's Reference*.

Verify the Mode

Before you make a backup, execute the following command to verify that your source database server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of your source database server. [Figure 8-2](#) shows that the database server is in quiescent mode.

```
INFORMIX-OnLine Dynamic Server  Version x.xx.xxx--Quiescent--Up  xx:xx:xx--xxxx
```

```
OnLine Dynamic Server is in quiescent mode.
```

```
Kbytes
```

Figure 8-2 Example of *onstat* Status Line

Back Up the Source Database Server

Use ON-Bar, **ontape**, or ON-Archive to make a level-0 backup and logical-log backup of Dynamic Server 7.3x or OnLine Dynamic Server 7.2x. If you use **ontape**, execute the following command to make a level-0 backup:

```
ontape -s
```

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. (For more information about making backups, refer to your *Archive and Backup Guide*.)

After you make a level-0 backup, perform a complete backup of the logical log, including the current logical-log file.

If you use ON-Archive, execute the following command to make a full-system, level-0 backup:

```
Onarchive> ARCHIVE/DBSPACESET=*
```

Change UNIX Kernel Parameters

You might need to change some of the UNIX kernel parameters before you install Extended Parallel Server. To reconfigure the operating system, follow the directions in the machine notes file included on your database server distribution media and the kernel-configuration instructions for your operating system.

For information on the location of the machine notes file, refer to [“Documentation Notes, Release Notes, and Machine Notes”](#) on page 22 of the Introduction.



Important: *Make sure you modify the kernel parameters on every node.*

Edit the Schema File to Suit Extended Parallel Server

Configure the downloaded schema for Extended Parallel Server. For information about schema configuration for Extended Parallel Server, refer to [“Database Schema”](#) on page 10-11.

Set Environment Variables

Before you install Extended Parallel Server or invoke DB-Access, set up the following environment variables on every node:

- **INFORMIXDIR**
- **INFORMIXSERVER**
- **ONCONFIG**
- **PATH**
- **INFORMIXSQLHOSTS**

The client application looks for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

Set the environment variable **PATH** so that the local directory that contains Informix commands is searched before the **INFORMIXDIR** directory.

The installation script installs Extended Parallel Server into the **INFORMIXDIR** directory specified for user **root** or **informix** on UNIX. The installation script does not bring Extended Parallel Server on-line.

Install Extended Parallel Server

To install and configure Extended Parallel Server, follow the directions in your *Installation Guide* and in your *Administrator's Guide*.

You must log in as user **root** or **informix** and set the **INFORMIXDIR** environment variable to the directory where you plan to install the database server. Install it on the node that contains the connection coserver in the directory that was NFS mounted on all the other nodes. Install the entire distribution on a single node within a file system that is shared across all the nodes that are assigned to Extended Parallel Server.

Check that the file system can hold the entire Extended Parallel Server distribution of approximately 180 megabytes. Export this file system with write access as user **root** or **informix** over the NFS and mount it to the same mount point on every node.



Warning: *If you install Extended Parallel Server in the same directory where Dynamic Server 7.3x or OnLine Dynamic Server 7.2x resides, the installation script overwrites the older files. If you want to transfer data directly to Extended Parallel Server from Dynamic Server 7.3 or OnLine Dynamic Server 7.2x, install Extended Parallel Server in a different directory. If you want to install Extended Parallel Server in the same directory, copy the Dynamic Server 7.3x or OnLine Dynamic Server 7.2x files to a different directory before you begin the installation.*

Before you overwrite the source database server, take the following precautions:

- If you do not have the original media for the source database server, back up the **INFORMIXDIR** directory before you install Extended Parallel Server.
- Copy the configuration files in the **etc** directory of **INFORMIXDIR** to another location on the file system.

The installation script installs Extended Parallel Server into the **INFORMIXDIR** directory specified for user **root**. The installation script does not bring Extended Parallel Server on-line.

Check the release notes for information about the correct operating-system patches, recommended shared-memory parameters, and configurations that are required for successful installation and operation of the database server. The machine notes and release notes are in the following directory:

\$INFORMIXDIR/release/en_us/0333 directory.

Make any changes that the machine notes and release notes recommend on every node. For more information on how to install Extended Parallel Server, refer to your *Installation Guide* and your *Administrator's Guide*.

Copy Database Server Utilities to Each Node

To ensure rapid and proper initialization, create local copies of the following utilities on each node:

- **oninit**
- **onmode**
- **onstat**

When you finish copying the utilities to each node, log out as **root**.

Update the ONCONFIG Configuration File

You can customize your **ONCONFIG** configuration file and environment variables to take advantage of the new features that Extended Parallel Server introduced. After you observe the performance of Extended Parallel Server, you might want to make further adjustments.

For information on how to configure Extended Parallel Server, refer to your *Administrator's Guide*. For information about environment variables, refer to the *Informix Guide to SQL: Reference*. For information about how to tune the configuration parameters, refer to your *Performance Guide*.

Bring Extended Parallel Server On-Line

Execute the following command to bring Extended Parallel Server on-line for the first time:

```
xctl -C oninit -iy
```

Check your database server message log to verify that all coservers are up. To view the message log, execute:

```
onstat -m
```

To verify that all your coservers are up and running, execute:

```
xctl onstat -
```

For information about messages in the message log, refer to your *Administrator's Guide*.



Important: If the message file notes problems, solve the problems before you continue to the next step.

Use onutil to Create Cogroups and Dbslices

Create cogroups and dbslices of equal sizes across all coservers. When you create a dbslice, you specify the cogroup name so that Extended Parallel Server knows the coservers on which to create dbspaces. For example, you might create a dbslice from an accounting cogroup. The following example shows how to create a cogroup and a dbslice:

```
% onutil
1> CREATE COGROUP acctg_cogroup
2> FROM xps.%r(1..8);
Cogroup successfully created.
3> CREATE DBSLICE acctg_dbslc
6> FROM COGROUP acctg_cogroup ...
```

You do not need to specify the names explicitly for all the individual dbspaces that are associated with the partitioned tables. Extended Parallel Server generates the dbspace names for you.

Run **xctl** to verify that all your dbspaces were created correctly on each coserver:

```
xctl onstat -d
```

Edit the onxfer Configuration File

You can configure each **onxfer** process by setting parameters in the configuration file before starting the utility. The default name of the **onxfer** configuration file is **xfer_config** in the **\$INFORMIXDIR/etc** directory. You can specify a different directory in the **XFER_CONFIG** environment variable.

For details about onxfer configuration parameters, see [Chapter 26, “Using the onxfer Utility.”](#)

Unload the 7.3x or 7.2x Database Schema and Data

Start the **onxfer** utility to unload the Version 7.3x or Version 7.2x schema and data. For details about **onxfer** syntax, see [Chapter 26, “Using the onxfer Utility.”](#)

The schema on the source database is downloaded into the directory that you specified. The directory contains the following three unloaded files:

- **db_pre.sql**
- **db_post.sql**
- **db_incompat.sql**

Load the Database Schema and Data

The schema is created based on the **db_pre.sql** and **db_post.sql** files. You need to check the **db_incompat.sql** file for parts of the schema that you might need to create after the initial loading.

Make an Initial Backup of Extended Parallel Server

Use the backup and restore tool (ON-Bar) to make a level-0 backup of Extended Parallel Server. Do not overwrite the tapes you used to make the final backup of Dynamic Server 7.3x or OnLine Dynamic Server 7.2x. For more information about ON-Bar, see your *Backup and Restore Guide*.

Extended Parallel Server supports unloading directly to a pipe or to a tape drive. You do not need to unload data first to disk.

You can also use external tables to perform a backup. For information about loading data into and unloading data from external tables, see your *Administrator's Guide*.

Tune Extended Parallel Server for Performance

When you finish the level-0 backup, the migration process is complete and users can use Extended Parallel Server to access data safely.

After you successfully migrate to Extended Parallel Server, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between the 7.x and 8.3 database servers. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on performance topics, refer to your *Performance Guide*.

Reverting to Dynamic Server 7.3x or OnLine Dynamic Server 7.2x from Extended Parallel Server

After you migrate to Extended Parallel Server from Dynamic Server 7.3x or OnLine Dynamic Server 7.2x, you cannot revert back automatically. To revert, you need to unload the data into an external table from the 8.3 database server and then load it back into the 7.x database server.

Use the **onutil** utility to check for outstanding in-place ALTER TABLE versions, as follows:

```
>onutil  
>CHECK TABLE INFO
```

If pages exist with old table definitions, update and upgrade all data pages to current table definitions. To do this, run a dummy update to change all data pages to the latest definition, as your *Performance Guide* describes.

Unloading Data to External Tables

To unload data to an external file, issue SELECT with INTO EXTERNAL statements. You can implicitly or explicitly specify an external table. (If you are loading data, an explicit definition is mandatory.) The SELECT with INTO EXTERNAL statements create a default external table description for unloading data. Issue the following SQL statement:

```
SELECT *
FROM customer
WHERE customerNum > 100
AND lastName [1] = "A"
INTO EXTERNAL TABLE custExtII
USING
(
  FORMAT "informix",
  DATAFILES
  ( "DISK:1:/tmp/dat.out",
    "DISK:2:/tmp/dat.out",
    "DISK:3:/tmp/dat.out"
  ));
```

In this example, the external table is implicitly defined when the table is unloaded. If the external table already exists, you can use the following syntax:

```
INSERT INTO extTableName
SELECT *
FROM tableName
WHERE...;
```

For more information about external tables, see the *Administrator's Reference*.

Loading Data from External Tables

After the data is unloaded into external tables, you can use the load utilities in Dynamic Server 7.3x or OnLine Dynamic Server 7.2x to load the data. For load-utility information, see [Chapter 2, "Data Migration."](#)

Migrating Between Dynamic Server AD/XP 8.21.UD4 and OnLine XPS 8.11

In This Chapter	9-3
Preparing for Migration	9-3
Migration Guidelines.	9-4
Changes in Dynamic Server AD/XP 8.21	9-4
Storage-Manager Validation and Installation	9-5
Upgrading to Dynamic Server AD/XP 8.21.UD4 from	
OnLine XPS 8.11	9-5
Check the Available Space and System Requirements	9-6
Save Copies of the Current Configuration Files.	9-6
Close All Transactions and Shut Down the 8.11	
Database Server	9-7
Install Dynamic Server AD/XP 8.21.UD4.	9-8
Make an Initial Backup of Dynamic Server	
AD/XP 8.21.UD4.	9-12
Moving Dynamic Server AD/XP 8.21.UD4 Between 32-Bit	
and 64-Bit Operating Systems	9-12
Reverting to an Earlier 8.x Database Server from	
Dynamic Server AD/XP 8.21.UD4	9-12

In This Chapter

This chapter describes how to migrate between Dynamic Server with AD and XP Options, Version 8.21.UD4, and OnLine XPS, Version 8.11. The following topics are in this chapter:

- Preparing for migration
- Upgrading to Dynamic Server with AD and XP Options 8.21.UD4 from OnLine XPS 8.11
- Moving Dynamic Server with AD and XP Options 8.21.UD4 between 32-bit and 64-bit operating systems
- Reverting to OnLine XPS 8.11 from Dynamic Server with AD and XP Options 8.21.UD4

Preparing for Migration

To prepare for migration between Dynamic Server with AD and XP Options 8.21.UD4 and OnLine XPS 8.11, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration.

Migration Guidelines

Informix recommends that you observe the following guidelines when you migrate between Dynamic Server with AD and XP Options 8.21.UD4 and OnLine XPS 8.11:

- Review the release notes and machine notes for information about the correct operating-system release, new 8.21 features, installation procedures, and fixes to problem. For information on features not covered in the manuals, check the documentation notes.

The release notes, machine notes, and documentation notes are in the following directory:

`$INFORMIXDIR/release/en_us/0333`

- Retain the installation media for both versions of the Informix product software.
- Before you upgrade to the target database server from the source database server, make sure that there are no open transactions in the source database server. Fast recovery will fail when rolling back open transactions during the upgrade.
- Before the migration procedure, perform a level-0 backup of all dbspaces of the source database server. After you complete the migration, perform another level-0 backup of the target database server.
- Use a test instance of the database server to test the installation and migration procedures.
- Verify storage-manager validation for the target database server.

For details, see [“Storage-Manager Validation and Installation” on page 9-5](#).

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Changes in Dynamic Server AD/XP 8.21

For information about new and changed features, see the release notes for Dynamic Server with AD and XP Options 8.21.UD4.

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the **ism_catalog** utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Upgrading to Dynamic Server AD/XP 8.21.UD4 from OnLine XPS 8.11

This section describes the procedures for upgrading to Dynamic Server with AD and XP Options, Version 8.21.UD4, from OnLine XPS, Version 8.11.

When you migrate to Dynamic Server with AD and XP Options 8.21.UD4, you can install and test a database server instance with the same configuration files, environment variables, and **sqlhosts** information that you used for your source database server. After you install Dynamic Server with AD and XP Options 8.21.UD4 and verify that it works, you might want to modify configuration files and environment variables to take advantage of the Dynamic Server with AD and XP Options 8.21.UD4 features. For more information, refer to your *Getting Started* manual, your *Performance Guide*, and your *Administrator's Guide*.

To upgrade to Dynamic Server with AD and XP Options 8.21.UD4, complete the following steps, which later sections describe in detail:

1. Install the latest maintenance release for the 8.11 database server.
2. Check the available space and system requirements.
3. Save copies of the current configuration files.
4. Close all transactions and shut down the 8.11 database server.
5. Install Dynamic Server with AD and XP Options 8.21.UD4.
6. Make an initial backup of the 8.21.UD4 database server.



Important: The upgrade procedure might not be restartable. In case of a failure, you might need to restore your source database server from a backup.

Check the Available Space and System Requirements

Dynamic Server with AD and XP Options 8.21.UD4 requires approximately 2,500 free pages (10 megabytes) of logical-log space to build the **sysmaster** database. Approximately 70 data pages (280 kilobytes) should be available for each database.

The catalogs are created in the same dbospace in which the database was created. If no dbospace was used to create the database, the root dbospace on coserver 1 is used as the default. If there is not sufficient free space in the affected dbospace, you should add a new chunk before you perform the upgrade.

For more information on disk-space requirements for Dynamic Server with AD and XP Options 8.21.UD4, refer to your *Administrator's Guide*.

Save Copies of the Current Configuration Files

Save copies of the configuration files and the **sqlhosts** information for each instance of Dynamic Server with AD and XP Options 8.21.UD4. Keep the copies available for later use. Save the configuration files that the following list shows:

- **\$INFORMIXDIR/etc/\$ONCONFIG**
- **\$INFORMIXDIR/etc/onconfig.xps**
- **\$INFORMIXDIR/etc/onconfig.std**

- `$INFORMIXDIR/etc/sqlhosts`
- `$INFORMIXDIR/etc/termcap`
- `$INFORMIXDIR/etc/oncfg*`
- `$INFORMIXDIR/etc/xcfg*`

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file from all nodes:

`$INFORMIXDIR/etc/Bixbar_*.<servernum>`

Informix recommends that you save the entire 8.11 installation directory tree. The following example shows an easy way to do this on UNIX:

```
cd $INFORMIXDIR
tar -cvf /mybackupdir/informix811.tar
```

Running this **tar** command creates a single file that contains all the files in the directory tree.

Close All Transactions and Shut Down the 8.11 Database Server

Before you migrate to the 8.21.UD4 database server, make sure that no connections exist to the source database server. Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and shut down the 8.11 database server.

Before you upgrade your 8.11 database server to Version 8.21.UD4, make sure that no open transactions exist. Fast recovery would fail when rolling back open transactions during the upgrade.

To let users exit and shut down the database server gracefully

1. Execute the following command to put the database server into quiescent mode:

```
xctl .do onmode -sy
```

2. Wait for all users to exit.

3. Execute the following command to force a checkpoint:

```
xctl .do onmode -c
```

4. Perform a level-0 backup of the database server.
5. Execute the following command to shut down the system:

```
xctl .do onmode -yuk
```

6. To guarantee that no open transactions exist, execute the following command on all coservers to put the database server in quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

The **oninit -s** command rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. For more information about fast recovery and quiescent mode, refer to your *Administrator's Guide*.

7. Issue the following command again to shut down the database server:

```
xctl .do onmode -yuk
```

8. To verify the mode of the database server, execute the following command:

```
xctl onstat -
```

9. Check the message log to see that all coservers are shut down.

Install Dynamic Server AD/XP 8.21.UD4

To install Dynamic Server with AD and XP Options 8.21.UD4 on UNIX, you must log in as user **root**. Set the **INFORMIXDIR** environment variable to the directory where you plan to install the 8.21.UD4 database server.



Warning: If you install the 8.21.UD4 database server in the same directory where the 8.11 database server resides, the installation script overwrites the older files. If you want to preserve your 8.11 files, copy them to a different directory and then install the 8.21.UD4 database server in the same directory.

Check the machine notes and release notes for information about the correct operating-system patches, recommended shared-memory parameters, and configurations that successful installation and operation of the database server requires. The release notes are in the following directory:

\$INFORMIXDIR/release/en_us/0333

Make any changes that the machine notes or release notes recommend on every node.

To install the 8.21.UD4 database server

1. Verify that the following environment variables are set correctly by reviewing the output of the **env** command on every node:
 - **INFORMIXDIR**
 - The installation script installs Dynamic Server with AD and XP Options into the **INFORMIXDIR** directory specified for user **root** or **informix**.
 - **INFORMIXSERVER**
 - **INFORMIXSQLHOSTS**

The client applications look for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.
 - **ONCONFIG**
 - **PATH**

Set the environment variable **PATH** so that the local directory is searched before the **INFORMIXDIR** directory.

For the correct settings of these environment variables, refer to your *Administrator's Guide*. You can change the values of these environment variables after you install Dynamic Server with AD and XP Options 8.21.UD4 and before you invoke DB-Access.
2. Install the 8.21.UD4 database server after you install the Informix Client Software Developer's Kit (Client SDK):
 - a. Change directory (**cd**) to the target directory and issue the following command or commands to untar the Client SDK product:

```
tar -svf file.tar
```

or

```
zcat file.tar.Z  
tar -xvf -
```
 - b. Install the Client SDK product (follow the installation instructions).

- c. Change directory (**cd**) to the target directory and issue the following command or commands to untar Dynamic Server with AD and XP Options 8.21.UD4:

```
tar -svf file.tar
```

or

```
zcat file.tar.Z  
tar -xvf -
```

- d. Install the database server as user **root** (follow the installation instructions).
3. Check your **PATH** environment variable by running the following command:

```
ls -l 'which oninit'
```

The execution path should find the new 8.21.UD4 **oninit**. If you are using NFS, copy the critical files (**oninit**, **onstat**, **onutil**, and **onmode**) to the local file system on all nodes and insure that the **PATH** environment variable points to the local files.

4. Bring the 8.21.UD4 database server on-line with the new product.

Warning: Do not use the **-i** flag.

Start the database server as you normally would. The database server will start the migration automatically. The migration process occurs only the first time the database server is started. This phase performs any disk conversion and rebuilds the **sysmaster** and **sysutils** databases.

This process might run for an extended period of time. The database server notes the progress in the **online.log** file. To monitor the progress, run the following command:

```
tail -f online.log
```

The database server puts the following message in the log as each dbspace conversion completes:

```
Conversion complete for dbspace dbspace_name.
```

The 8.21.UD4 initialization upgrades up to 20 dbspaces in parallel. To verify that the upgrade is in process, you can use the **iostat** command. Significant I/O occurs on a disk device during conversion of a dbspace.



5. Verify that the **sysmaster** and **sysutils** databases built successfully and that all coservers are on-line.

If you are upgrading to Dynamic Server with AD and XP Options 8.21.UD4 from an 8.11 database server earlier than 8.11.UF1, and you did not restart the database server with **oninit -iy**, drop and re-create the **sysutils** database. A new column has been added to a table in the **sysutils** database. Log in as user **informix** or **root** and execute **\$INFORMIXDIR/etc/bldutil.sh**. The **bldutil.sh** command destroys all backup history.

For more information about the **sysmaster** and **sysutils** databases, see your *Administrator's Guide* and your *Backup and Restore Guide*.

6. Access every database in the server sequentially using the following command in DB-Access:

```
database database_name;
```

This phase performs the final upgrade steps for each database in the server.

7. Perform a level-0 backup of the newly upgraded database server instance.
8. Shut down the database server and then bring it back on-line.
9. Set any new environment variables that you might need for your database server prior to restarting it.

Informix recommends the following settings for the ONCONFIG parameters **LBUFFER** and **SBUFFER** to achieve better performance:

- **LBUFFER = 12288**
- **SBUFFER = 2048**

For detailed information on these parameters, refer to the release notes.

Make an Initial Backup of Dynamic Server AD/XP 8.21.UD4

Use the backup and restore tool (ON-Bar) to make a level-0 backup of Dynamic Server with AD and XP Options 8.21.UD4. For more information about ON-Bar, see your *Backup and Restore Guide*.

Instead of ON-Bar, you can use external tables to perform a backup. For information about loading data to and unloading data from external tables, see the *Administrator's Reference*.

Moving Dynamic Server AD/XP 8.21.UD4 Between 32-Bit and 64-Bit Operating Systems

To move Dynamic Server with AD and XP Options 8.21.UD4 from a 32-bit operating system to a 64-bit operating system, shut down the database server cleanly and bring it back up on the 64-bit operating system. Similarly, to move Dynamic Server with AD and XP Options 8.21.UD4 from a 64-bit operating system to a 32-bit operating system, shut down the database server cleanly and bring it back up on the 32-bit operating system.

Reverting to an Earlier 8.x Database Server from Dynamic Server AD/XP 8.21.UD4

After you migrate to Dynamic Server with AD and XP Options 8.21.UD4 from an earlier 8.x database server, you cannot revert back automatically. To revert, you need to unload the data into an external table from the 8.21.UD4 database server and then load it back into the earlier 8.x database server.

Migrating Between Dynamic Server AD/XP 8.21 and OnLine Dynamic Server 7.2

In This Chapter	10-3
Preparing for Migration	10-4
Changes That Dynamic Server AD/XP 8.21 Introduced	10-5
Environment Variables	10-5
Configuration Parameters	10-5
The sqlhosts File or Registry Key	10-8
Database Schema	10-11
Logging and Nonlogging Table Types	10-13
Utilities	10-15
Informix Enterprise Command Center	10-17
New Indexes	10-17
Global Language Support	10-18
Backup Tools and ON-Bar Configuration	10-18
Storage-Manager Validation and Installation	10-18
Loading and Unloading Data	10-19
Express Mode	10-19
Deluxe Mode	10-20
Upgrading to Dynamic Server AD/XP 8.21 from OnLine	
Dynamic Server 7.2x.	10-20
Configure and Check Available Space	10-23
Save Copies of the Current Configuration Files.	10-24
Close All Transactions in the Source Database Server.	10-25
Put OnLine Dynamic Server 7.2x in Quiescent Mode.	10-25
Verify the Integrity of the Data	10-26
Verify the Mode	10-27
Make a Final Backup of OnLine Dynamic Server 7.2x	10-27
Unload the Data	10-28
Take OnLine Dynamic Server 7.2x Off-Line	10-28

Change UNIX Kernel Parameters.	10-29
Install Dynamic Server AD/XP 8.21.	10-29
Set Environment Variables	10-30
Copy Database Server Utilities to Each Node	10-31
Update the ONCONFIG Configuration File	10-31
Update the sqlhosts File or Registry Key	10-31
Update the Backup and Restore Configuration Parameters	10-31
Bring Dynamic Server AD/XP 8.21 On-Line	10-32
Use onutil to Create Cogroups and Dbslices	10-32
Create a Database	10-33
Load Data from External Tables	10-33
Specify DATAFILES	10-34
Set Log File and Maximum Errors	10-35
Run UPDATE STATISTICS and Build Indexes	10-36
Verify the Integrity of the Data	10-36
Make an Initial Backup of Dynamic Server AD/XP 8.21	10-37
Check Rejection Files	10-37
Recompile ESQL/C Programs.	10-38
Complete Migration	10-38
Reverting to OnLine Dynamic Server 7.2x from Dynamic	
Server AD/XP 8.21	10-38
Unloading Data to External Tables	10-39
Loading Data from External Tables	10-39

In This Chapter

This chapter describes the procedure to migrate between Dynamic Server with AD and XP Options, Version 8.21, and OnLine Dynamic Server, Version 7.2x.

This chapter covers the following topics:

- Preparing for migration
- Loading and unloading data
- Upgrading to Dynamic Server with AD and XP Options 8.21 from OnLine Dynamic Server 7.2x
- Reverting to OnLine Dynamic Server 7.2x from Dynamic Server with AD and XP Options 8.21

***Tip:** To migrate to Dynamic Server with AD and XP Options 8.21 from Dynamic Server 7.3x, revert to OnLine Dynamic Server 7.2x first.*



UNIX

WIN NT

Preparing for Migration

To prepare for migration between Dynamic Server with AD and XP Options 8.21 and OnLine Dynamic Server 7.2x, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration.

Informix suggests that you observe the following precautions when you migrate to Dynamic Server with AD and XP Options 8.21:

- Check the release notes for information about the proper operating-system release and any patches that you require for successful installation and operation of the database server.

The release notes are in one of the following directories:

- `$INFORMIXDIR/release/en_us/0333` ♦

- `%INFORMIXDIR%\release\en_us\0333`.

Release notes appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar. ♦

On UNIX, retain both versions of the Informix product software on disk (if you have enough disk resources). On Windows NT, you cannot retain two versions of the Informix product on disk.

- Retain the installation media from both versions of the Informix product software.
- Perform a level-0 backup of all dbspaces and blobspaces with OnLine Dynamic Server 7.2x.

After you complete the migration, perform another level-0 backup with Dynamic Server with AD and XP Options 8.21.

- Use a test instance of Dynamic Server with AD and XP Options 8.21 to test the installation and migration procedures.

Use a test instance in the desired communications mode to practice bringing the new database server on-line before you attempt to upgrade the production database.

- Verify storage-manager validation for the target database server.

For details, see, [“Storage-Manager Validation and Installation” on page 10-18](#).

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Changes That Dynamic Server AD/XP 8.21 Introduced

This section describes the changes that Dynamic Server with AD and XP Options 8.21 introduced that affect initial configuration and migration. Dynamic Server with AD and XP Options introduced new features and associated terminology, including coservers, cogroups, and dbslices that facilitate a high degree of parallelism.

Environment Variables

Dynamic Server with AD and XP Options 8.21 introduced significant changes to the following environment variables:

- **INFORMIXSERVER**
- **PDQPRIORITY**
- **PSORT_NPROCS**

The *Informix Guide to SQL: Reference* describes these environment variables. For performance implications and guidelines, refer to the *Performance Guide for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options*.

Configuration Parameters

Dynamic Server with AD and XP Options 8.21 introduced new configuration parameters that might affect installation. You might also need to adjust the values of existing parameters. The following figures list the parameters. For more information on these configuration parameters, see your *Administrator's Guide* or the *Administrator's Reference*.

[Figure 10-1](#) lists the ON-Bar and ISM parameters for a database server.

Figure 10-1
ON-Bar and ISM Configuration Parameters for a Database Server

Parameters Added	Parameters Changed	Parameters Dropped
BAR_ACT_LOG		
BAR_BSALIB_PATH		
BAR_DBS_COSVR		
BAR_LOG_COSVR		
BAR_RETRY		
BAR_SM		
BAR_SM_NAME		
BAR_WORKER_COSVR		
BAR_WORKER_MAX		
BAR_XFER_BUFSIZE		
BAR_XPORT_COUNT		
ISM_LOG_POOL		

Single Coserver

For a single coserver configuration, use the default values in the **onconfig.std** template in the **etc** directory in **INFORMIXDIR**.



Important: Do not modify **onconfig.std**. The database server provides the **onconfig.std** configuration file as a template and not as a functional configuration.

Figure 10-2 lists the ONCONFIG parameters for a single coserver configuration for Dynamic Server with AD and XP Options 8.21.

Figure 10-2
Single Coserver Configuration Parameters

Parameters Added	Parameters Changed	Parameters Dropped
CONFIGSIZE	MIRRORPATH	DS_MAX_SCANS
PHYSDBS	ROOTNAME	
ROOTNAME	ROOTPATH	
ISO_CURLOCKS	DBTEMPSPACE	
DS_ADM_POLICY	DS_TOTAL_MEMORY	
BAR_ACT_LOG	MAX_PDQPRIORITY	
BAR_RETRY	DBSERVERNAME	
BAR_WORKER_MAX		
BAR_WORKER_MAX		
BAR_BSALIB_PATH		
BAR_XPORT_COUNT		
BAR_XFER_BUFSIZE		
LOG_BACKUP_MODE		
LOG_BACKUP_MODE		
ISM_LOG_POOL		
BAR_ACT_LOG		
BAR_RETRY		
BAR_WORKER_MAX		



Multiple Coservers

For a multiple coserver configuration, use the default values in the **onconfig.xps** template that is located in the **etc** directory in **INFORMIXDIR**.

Important: Do not modify **onconfig.xps**. The database server provides the **onconfig.xps** configuration file as a template and not as a functional configuration.

Figure 10-3 lists the ONCONFIG parameters for a multiple coserver configuration for Dynamic Server with AD and XP Options 8.21.

Figure 10-3
Multiple Coserver Configuration Parameters

Parameters Added	Parameters Changed	Parameters Dropped
CONFIGSIZE	MIRRORPATH	DS_MAX_SCANS
COSERVER	ROOTNAME	
END	ROOTPATH	
PHYSSLICE	DBTEMPSPACE	
NODE	DS_TOTAL_MEMORY	
ROOTSLICE	MAX_PDQPRIORITY	
ISO_CURLOCKS	DBSERVERNAME	
DS_ADM_POLICY		
LOG_BACKUP_MODE		
LOG_BACKUP_MODE		

The sqlhosts File or Registry Key

Dynamic Server with AD and XP Options 8.21 contains formatting changes and new options in the **sqlhosts** file or registry key. The **dbservername** format and the options within the options field are different from any prior version.

The dbservername Identifier

Within the **sqlhosts** file or registry key for Dynamic Server with AD and XP Options, the **dbservername** identifies each connection coserver uniquely by the following format:

dbservername.coserver-number

Element	Description
<i>dbservername</i>	The value that you specify in the DBSERVERNAME or DBSERVERALIASES configuration parameter
<i>coserver-number</i>	The integer that you specify in each COSERVER configuration parameter

This form of the dbservername is referred to as a *coserver name*. Dynamic Server with AD and XP Options uses the DBSERVERNAME and coserver numbers specified in the **ONCONFIG** file to generate the following coserver names automatically:

```
xps.1
:
:
xps.8
```

New Options Syntax

The fifth field in the **sqlhosts** or registry fields, the **options** field, accommodates new options and future options.

The following list is a review of the **sqlhosts** or registry key fields:

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5
DBSERVERNAME	NETTYPE	HOSTNAME	SERVICENAME	OPTIONS

The **dbservername** in field 1 is a key for connectivity information in the remaining fields in the **sqlhosts** file or registry key.

The **options** field contains columns. Each column is separated by a comma or white space that represents an end of the column. Client and database server applications check each column to determine whether the option is supported in the database server release.



Tip: If you maintain more than one version of the database server, use separate **sqlhosts** files or registry entries for older versions of the database server. Alternatively, you can use separate entries with an alias to the appropriate database server.

For Dynamic Server with AD and XP Options 8.21, an **sqlhosts** entry must exist for every coserver. The **sqlhosts** file or registry key contains a line for each connection type that the database server provides, for each coserver that makes up Dynamic Server with AD and XP Options, and for each 8.21 database server to which a client connects.

Figure 10-4 lists the **sqlhosts** file or registry key fields for Dynamic Server with AD and XP Options 8.21.

Figure 10-4
sqlhosts File or Registry Fields

dbservername	nettype	hostname	servicename	options
dbservername.coserver_1	onsoctcp	node1	sqlexec.1	e=x100,r=1,s=1
dbservername.coserver_2	onsoctcp	node2	sqlexec.2	k=1,r=1, b=200
⋮	⋮	⋮	⋮	⋮
dbservername.coserver_8	onsoctcp	node8	sqlexec.8	g=abcd,i=2300

In the sample **sqlhosts** file, the **options** field for the first coserver contains three options in three columns, as the following table shows.

Column	Option
Column 1	e=x100
Column 2	r=1
Column 3	s=1



Important: Informix recommends that you use field 5, **options**, for Dynamic Server with AD and XP Options 8.21 for the following options only: **b, k, r, s, g, e, c, i**. Place any other options in subsequent columns. If you do not want any of these options but do want other options, use **k=1** in column 5, which is the default. For example, for options for **dbservername.cosever_8**, use **k=1, g=abcd, i=2300**.

For more information on the components of the **sqlhosts** file or registry key, see the *Administrator's Guide for Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options*.

Database Schema

The Version 8.21 database schema supports flexible fragmentation strategies, system-defined hash fragmentation, hybrid fragmentation, and unique serial columns across coservers. When you migrate to Dynamic Server with AD and XP Options, make sure to configure your database schema to take advantage of these features, which allow for a high degree of parallelism.

Hybrid Fragmentation

Hybrid fragmentation is a schema that incorporates multiple fragmentation strategies. Prior to Dynamic Server with AD and XP Options 8.21, table-fragmentation strategy was based on a single strategy, either hash, expression, or round-robin. Dynamic Server with AD and XP Options 8.21 allows fragmentation by system-defined hash and you can combine two fragmentation strategies on the same table.

When you prepare to migrate to Dynamic Server with AD and XP Options 8.21, define the dbspace or dbspaces where your fragmented tables reside, as the following example shows:

```
CREATE TABLE account
(account_id integer,
 account_bal integer,
 account_date date,
 account_name char(30)
)
FRAGMENT BY HASH(account_date)
IN account_dbsp1,
   account_dbsp2,
   :
   :
account_dbspn;
```

This CREATE TABLE statement fragments the table across multiple dbspaces, which can reside on different coservers. You can create dbslices to manage the multiple dbspaces across coservers. The dbslices simplify administration of multiple dbspaces that reside on different coservers. For example, you can create a **dbslice account_dbslc** that includes **dbspaces account_dbsp1 to account_dbspn**.

The following statement shows how the CREATE TABLE statement is simpler with the IN dbslice clause:

```
CREATE TABLE account
(account_id integer,
 account_bal integer,
 account_date date,
 account_name char(30)
) FRAGMENT BY HASH(account_date)
  IN account_dbslc;
```

For more information on how to create a dbslice, refer to your *Administrator's Guide*.

Hybrid fragmentation allows you to further fragment the table within each dbslice. You might want to fragment each time period by account number, as the following sample CREATE TABLE statement shows:

```
CREATE TABLE account
(account_id integer,
 account_bal integer,
 account_date date,
 account_name char(30)
)
FRAGMENT BY HYBRID (HASH (account_date)) EXPRESSION
  account_id < 100 IN account_dbslc.1
  account_id >= 100 AND account_id > 200 IN account_dbslc.2
  :
  <expression_n> in account_dbslc.n;
```

For more information on hybrid fragmentation syntax, see the *Informix Guide to SQL: Syntax*. For more information and performance implications of using various fragmentation strategies, see your *Performance Guide*.

Unique Record Identifiers: Serial Columns

Serial columns in Dynamic Server with AD and XP Options are unique identifiers across coservers. In OnLine Dynamic Server, ROWIDs uniquely identify records, but Dynamic Server with AD and XP Options 8.21 does not use ROWIDs across multiple coservers as unique record identifiers. You can use ROWIDs in Dynamic Server with AD and XP Options, but they act as unique identifiers only if your tables are not fragmented.

Tip: When you migrate to Dynamic Server with AD and XP Options 8.21, make sure your applications do not assume that ROWIDs are unique.



Dynamic Server with AD and XP Options 8.21 uses serial columns that are contiguous within each coserver to serve as unique identifiers, although a gap in numbers between coservers might occur. You can load values into serial columns, such as from the original data file, or use values that Dynamic Server with AD and XP Options generates automatically. However, when you create a table, you must specify that you want serial columns.

Design your database schema to take advantage of fragmentation strategies and serial columns before you migrate to it. For general information about fragmentation and serial columns, see the *Informix Guide to Database Design and Implementation*.

Logging and Nonlogging Table Types

In Dynamic Server with AD and XP Options 8.21, no method exists to create a nonlogging database. The default logging mode is set to unbuffered logging. This database server, however, offers the ability to have logging and nonlogging *tables*, and you can mix logging and nonlogging tables within the same database.

Dynamic Server with AD and XP Options 8.21 has *permanent* and *temporary* tables with logging or nonlogging capabilities. Permanent tables have four classes: *raw*, *standard*, *operational*, and *static*. Temporary tables have two classes: *scratch* and *temp*. Both permanent and temporary tables can be altered from one class to another.

The logging mode of a table restricts the type of data load you can use for a table as well as the recoverability of the table.



Tip: Use raw tables to initially load and scrub data and later alter the tables to another class, if necessary. Raw tables can be loaded in express mode only. For a description of express-load mode, see [“Express Mode” on page 10-19](#).

The following table lists the logging and loading characteristics of permanent and temporary table types. Express- and deluxe-loading modes are discussed in [“Loading and Unloading Data” on page 10-19](#).

Table Type		Logging or Nonlogging	Load Mode Options
Permanent	Raw	Nonlogging	Express
	Standard	Logging	Deluxe
	Operational	Logging	Express or deluxe
	Static	Nonlogging	No loading
Temporary	Scratch	Nonlogging	
	Temp	Logging	

Permanent Tables

Use raw tables to initially load and scrub data. Raw tables use light appends that bypass the buffer cache and allow fast loading. They do not support indexes, referential constraints, rollback and recovery, or restoration from backup. You can use raw tables only with express-mode loads. For a description of express-load mode, see [“Express Mode” on page 10-19](#).

Standard tables are similar to tables in a logged database in OnLine Dynamic Server. They do not use light appends but support recovery and rollback. Standard tables allow restores. However, you cannot use express mode to load standard tables.

Use operational tables to load data from a source outside the database system. Operational tables use light appends—unbuffered, unlogged insert operations. They allow rollback and recovery but do not allow restoration from backup. If indexes are enabled, deluxe-mode load is automatic. For descriptions, see [“Deluxe Mode” on page 10-20](#).

Use static tables for read-only operations. Static tables use light scans and do not need locking. They allow constraints and nonclustered indexes but do not allow rollback and recovery or restoration from backup. Static tables permit advanced indexing methods created especially for decision-support system (DSS) queries.

Static tables do not support data manipulation operations (inserts, updates, and deletes) and they cannot be used for loading. To load the tables, alter the table type to load and then change the table type back to static.

Temporary Tables

Scratch tables are unlogged and use light appends. They do not support indexes, referential constraints, or rollback.

Temporary tables are logged and support bulk operations, including light appends. Temporary tables also support indexes, referential constraints, and rollback but do not support recovery.

For information on correct SQL statement syntax and use when you create and change tables, see the *Informix Guide to SQL: Syntax*. For more details about table characteristics, see your *Administrator's Guide*. For more information on how to work with table types and fragmentation to design your database, see the *Informix Guide to Database Design and Implementation*.

Utilities

Dynamic Server with AD and XP Options uses new command-line utilities, **onutil** and **xctl**. The **onutil** utility consolidates much of the functionality of **oncheck**, **onparams**, and **onspaces** into one command. (Dynamic Server with AD and XP Options does not support the **oncheck** utility.)

For a multiple coserver configuration, the **xctl** utility lets you execute command utilities and operating-system commands on one or more coservers. Informix recommends that you use **onutil** in Dynamic Server with AD and XP Options rather than other command utilities.

onutil

For a single coserver, use **onutil** to define or modify the following objects in Dynamic Server with AD and XP Options:

- Cogroups
- Dbslices
- Dbspaces
- Logical logs
- Logslashes (an object name for a collection of logs)



Tip: The **onutil** utility replaces the functionality of the **-cd**, **-pd**, **-pt**, **-pT**, **-pr**, **-pd**, **-cl**, **-ci**, **-pk**, **-pp**, **-pP**, and **-cc** options of **oncheck**.

To define or modify physical logs, use **onparams**.

Figure 10-5 compares the **oncheck** commands to the **onutil** equivalent.

oncheck Command	onutil Command
-cd, -pd	CHECK TABLE DATA
-pt, -pT	CHECK TABLE INFO
-cr, -pr	CHECK RESERVED
-cl	CHECK INDEX WITH DATA
-ci, -pk	CHECK INDEX
-pp, -pP	DISPLAY PAGE
-cc	CHECK CATALOGS
-ce, -pe	CHECK SPACE

Figure 10-5
*Comparable oncheck and
onutil Commands*

WIN NT

The installation procedure prepares a file, **setenv.cmd**, that sets the environment variables to their correct values. The **setenv.cmd** file is stored in the %INFORMIXDIR% directory. You must execute **setenv.cmd** before you can use any of the command-line utilities. You can execute the file automatically from Informix Enterprise Command Center (IECC). You can also execute **setenv.cmd** from the command line. ♦

xctl

For a multiple coserver configuration, you can use **xctl** with **onstat**, **oninit**, and **onmode** to execute command utilities and operating-system commands on a per-coserver basis.

For example, use **xctl** to initiate Dynamic Server with AD and XP Options on all your coservers:

```
xctl -C oninit -y
```

Several new options to **onstat** are available to check for OnLine XPS statistics. Use the following command to view the new **onstat** options:

```
xctl -onstat --
```

The following command brings down all coservers on a Dynamic Server with AD and XP Options instance:

```
xctl onmode -ky
```

For more information on **onutil** and **xctl**, refer to your *Administrator's Guide*.

WIN NT

Informix Enterprise Command Center

IECC for Dynamic Server with AD and XP Options runs on Windows NT. It enables the database server administrator to configure, control, and monitor the status of Dynamic Server with AD and XP Options. IECC provides a graphical interface that simplifies the process of database server administration and automates common database server administration functions.

IECC includes on-line help files. For additional information about IECC, see the *Informix Enterprise Command Center User Guide*.

New Indexes

Dynamic Server with AD and XP Options 8.21 supports *generalized-key* (GK) and *bitmap* indexes. GK indexes allow you to create indexes with any of the following key values:

- Subset of rows from a table
- Derived from an expression
- Join of columns from multiple tables
- Combination of various indexes on a table

Bitmap indexes can store a list of record identifiers (ROWIDS) for key values in a compressed bitmap format. GK indexes can be bitmap indexes as well.

When you upgrade to Dynamic Server with AD and XP Options 8.21, system catalog tables are added automatically to support GK indexes. However, because GK indexes are not supported in earlier versions of the database server, if you revert to an earlier version of the database server, drop GK indexes.

GLS

When you migrate to Dynamic Server with AD and XP Options 8.21, bitmap indexes that you create with the USING BITMAP syntax are not accessible by older versions of the database server. If you revert to an earlier version of Dynamic Server with AD and XP Options, drop bitmap indexes.

For more information on GK indexes and bitmap indexes, refer to your *Administrator's Guide* and your *Performance Guide*. For strategic use of different index types, refer to the *Informix Guide to Database Design and Implementation*.

Global Language Support

Dynamic Server with AD and XP Options incorporates Global Language Support (GLS). GLS enables Dynamic Server with AD and XP Options to handle different languages, cultural conventions, and code sets for Asian, European, Latin American, and Middle Eastern countries. The *Informix Guide to GLS Functionality* provides a full description of GLS.

Backup Tools and ON-Bar Configuration

Dynamic Server with AD and XP Options does not support the **ontape** utility or ON-Archive. To back up a Version 8.21 database server, you need to use ON-Bar or data migration tools.

If you use ON-Bar, when you migrate to Dynamic Server with AD and XP Options 8.21 from OnLine Dynamic Server 7.2x, you need to make substantial changes in the ON-Bar configuration. For details about configuring ON-Bar, refer to your *Backup and Restore Guide*.

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.



When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.

Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the **ism_catalog** utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Loading and Unloading Data

Dynamic Server with AD and XP Options 8.21 uses high-performance parallel loading with *external tables* to load and unload data. You can load tables from different sources and perform a variety of data-format conversions with external tables. The database server uses multiple threads to read data files in parallel and then convert the data into internal Informix format. A round-robin algorithm distributes the database server to the conversion threads.

You can use the default delimiter (|) format to load and unload files with INSERT... SELECT statements. You can also use other formats. For loading instructions, see [“Create a Database” on page 10-33](#).

High-performance parallel loading in Dynamic Server with AD and XP Options uses two modes: *express* and *deluxe*.

Express Mode

Express mode provides the highest performance during loads because it uses light appends. Light-append operations append rows to a table and bypass the buffer cache that eliminates buffer-management overhead. The table is locked exclusively during an express load so that no other user can access the table during the load. You cannot use express mode on a table with BYTE and TEXT data.

Use express mode on tables that do not have active indexes or referential constraints. Express mode is the default mode.



Warning: Data is not logged during an express-mode load, so you cannot recover data automatically.

Deluxe Mode

Deluxe mode combines fast parallel loading with an evaluation of indexes and unique constraints on a per-row basis. Deluxe mode is preferred if the cost of rebuilding an index is too high for the amount of data loaded. Data is logged during the load. If errors occur during a deluxe load, the load fails.

You can lock the table in shared mode to allow concurrent access by users, or alternatively, you can lock the table exclusively until the load is complete. Use deluxe mode if indexes, triggers, and referential constraints are enabled.

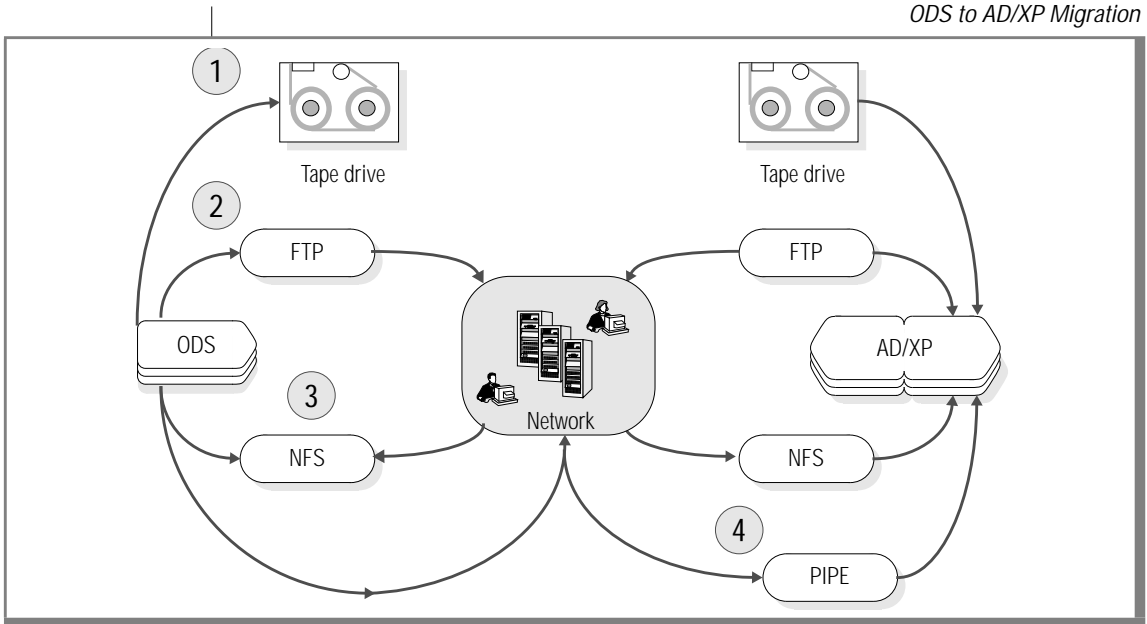
For complete documentation on external tables and deluxe- and express-mode loading syntax, see your *Administrator's Guide* and the *Informix Guide to SQL: Syntax*.

Upgrading to Dynamic Server AD/XP 8.21 from OnLine Dynamic Server 7.2x

You can use four options to migrate data to Dynamic Server with AD and XP Options 8.21 from OnLine Dynamic Server 7.2x. Your data-migration options depend primarily on connectivity between the source and target database servers and on the amount of disk space available for staging files.

[Figure 10-6](#) illustrates the four options for migrating data to Dynamic Server with AD and XP Options 8.21 from OnLine Dynamic Server.

Figure 10-6
ODS to AD/XP Migration



- **Option 1.** To migrate from OnLine Dynamic Server when no connectivity is available between source and target database servers, unload tables to tape before you load to Dynamic Server with AD and XP Options.
- **Option 2.** To migrate when the required disk space for staging files is available on *both* the source database server *and* the target database server, unload data from the source database to file.
Use file transfer protocol (FTP) to transfer the file to Dynamic Server with AD and XP Options to load.
- **Option 3.** To migrate when you have sufficient disk space for staging files *only* on Dynamic Server with AD and XP Options or the source database, use network file system (NFS) mount points.
For example, if Dynamic Server with AD and XP Options has enough disk space to handle your staging files, you can NFS mount a directory from Dynamic Server with AD and XP Options to OnLine Dynamic Server and unload your data from OnLine Dynamic Server to that NFS mounted directory.
- **Option 4.** You can use named pipes to support loading data to and unloading data from storage devices, including tape drives and direct network connections to mainframes.

UNIX



To upgrade data when you load, you can create a filter program that writes upgraded data to a named pipe. The database server then reads its input from the named pipe in one of the common formats. ♦

***Tip:** Option 4 is the fastest. Options 2 and 3 are faster than option 1 because of the time required to read and write to tape.*

This section describes the procedure for migrating to Dynamic Server with AD and XP Options, Version 8.21, from OnLine Dynamic Server, Version 7.2x. When you migrate, complete the following steps, which the sections that follow describe in more detail:

1. Configure and check available space.
2. Save copies of the current configuration files.
3. Close all transactions in the source database server.
4. Go to quiescent mode.
5. Verify the integrity of the data.
6. Verify the mode.
7. Make a final (level-0) backup.
8. Unload the data.
9. Take the source database server off-line.



***Important:** Repeat steps 1 through 9 for each instance of OnLine Dynamic Server that you are migrating to Dynamic Server with AD and XP Options 8.21.*

10. Reconfigure the operating system, if necessary.
11. Install Dynamic Server with AD and XP Options and copy **INFORMIXDIR** to each node.
12. Set environment variables on each node.
13. Copy database server utilities to each node.
14. Update the **ONCONFIG** configuration files.
15. Update the **sqlhosts** file or registry key.
16. Update the backup and restore configuration parameters.
17. Bring Dynamic Server with AD and XP Options on-line.
18. Use **onutil** to create cogroups and dbslices.
19. Create a database instance.
20. Load data from external tables.



21. Run UPDATE STATISTICS and build indexes.
22. Verify the integrity of the data.
23. Make an initial backup of Dynamic Server with AD and XP Options 8.21.
24. Check rejection files for each table you load.
25. Recompile ESQ/L/C programs.
26. Complete migration.

Important: Repeat steps 11 through 23 for each instance of Dynamic Server with AD and XP Options.

Configure and Check Available Space

Configure your computer memory equally among all the nodes. Because only one ONCONFIG file exists in Dynamic Server with AD and XP Options 8.21 and the configuration parameter is configured globally, the amount of memory you configure in your ONCONFIG file is limited to the node with the smallest amount of memory. If your system memory is configured unequally, Dynamic Server with AD and XP Options 8.21 cannot take advantage of the nodes that have extra memory.

When you calculate disk-space requirements for Dynamic Server with AD and XP Options 8.21, take into account the size of your tables. You need additional space for control information for your tables (approximately 60 bytes per page). Also consider disk space for the following needs:

- Root dbspace for all your nodes
- Physical logs
- Logical-log space
(If you require logging during loads, add the additional space.)
- Temporary dbspace
- Indexes
- Summary tables and index from your on-line transaction processing (OLTP) tools
- Mirroring
- Future growth
- File-system space for your operating system and Informix products

UNIX



- Staging space for loading and unloading files
- Safe write area
- New system catalog tables
- New **sysmaster** tables

Make sure you modify the kernel parameter on every node. ♦

For information on disk-space requirements for Dynamic Server with AD and XP Options, refer to your *Administrator's Guide*.

Important: *If a dbspace has less free space available than it requires, either move a table from the dbspace to another dbspace or add a chunk to it. The dbspace estimates could be higher if you have an unusually large number of SPL routines or indexes in the database.*

Save Copies of the Current Configuration Files

Save copies of the configuration files, if they are present, for each instance of OnLine Dynamic Server. Keep the copies available for later use. [Figure 10-7](#) lists the configuration files for each operating system.

Figure 10-7
ODS Configuration Files

UNIX	Windows NT
\$INFORMIXDIR/etc/\$ONCONFIG	%INFORMIXDIR%\etc\ONCONFIG
\$INFORMIXDIR/etc/onconfig	%INFORMIXDIR%\etc\onconfig\
\$INFORMIXDIR/etc/onconfig.std	%INFORMIXDIR%\etc\onconfig.std
\$INFORMIXDIR/etc/sm_versions	%INFORMIXDIR%\etc\sm_versions
\$INFORMIXDIR/aaodir/adtcfg	%INFORMIXDIR%\aaodir\adtcfg.*
\$INFOMIXDIR/dbssodir/adtmasks	%INFORMIXDIR%\dbssodir\adtmasks.*
\$INFORMIXDIR/etc/sqlhosts	
\$INFORMIXDIR/etc/tctermcap	
\$INFORMIXDIR/etc/termcap	

UNIX

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

UNIX: **\$INFORMIXDIR/etc/ixbar.<servername>**

Windows NT: **\$INFORMIXDIR\etc\ixbar.<servername>**

If you use ON-Archive to back up OnLine Dynamic Server 7.2x and logical logs, you must also copy and save the configuration files in the following list:

- **\$INFORMIXDIR/etc/SARC_CONFIG**
- **\$INFORMIXDIR/etc/config.arc**
- **\$INFORMIXDIR/etc/oper_deflt.arc ♦**

WIN NT

For Windows NT you do not need to copy these files because the Windows NT version of the 7.2x database server does not use ON-Archive. ♦

Close All Transactions in the Source Database Server

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and shut down the source database server. This procedure lets users exit and shuts down the system gracefully. If necessary, you can perform an immediate shutdown.

To let users exit and shut down the system gracefully

1. Execute the **onmode -sy** command.
2. Wait for all users to exit.
3. Execute the **onmode -ky** command.

To perform an immediate shutdown

```
onmode -ky
```

Put OnLine Dynamic Server 7.2x in Quiescent Mode

Execute the following command to enter quiescent mode and initiate a fast recovery of your current database:

```
oninit -s
```

The **oninit -s** option rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending. (For more information about fast recovery and quiescent mode, refer to your *Administrator's Guide*.)

You must execute **oninit -s** before you initialize your target database server. If the system is not left in a quiescent state, you receive the following error when you attempt to initialize Dynamic Server with AD and XP Options and it goes off-line:

```
Open transaction detected when changing log versions.
```

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data before you make a level-0 backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes.

To obtain the database names, use the following statements with DB-Access:

```
DATABASE sysmaster;  
SELECT name FROM sysdatabases;
```

Figure 10-8 lists the **oncheck** commands you can use to verify data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 10-8
*Commands for
Verifying the Data
Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

Verify the Mode

Before you make a backup, execute the following command to verify that your database server is in quiescent mode:

```
onstat -
```

The first line of the **onstat** output contains the status of the database server. [Figure 10-9](#) shows that the database server is in quiescent mode.

```
INFORMIX-OnLine Dynamic Server  Version x.xx.xxx  --  Quiescent  --  Up  xx:xx:xx
OnLine Dynamic Server is in quiescent mode.
- xxxx Kbytes
```

Figure 10-9
*Example of onstat
Status Line*

Make a Final Backup of OnLine Dynamic Server 7.2x

Use ON-Bar, **ontape**, or ON-Archive to make a level-0 backup and logical-log backup of OnLine Dynamic Server. If you use **ontape**, execute the following command to make a level-0 backup:

```
ontape -s
```

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup.

After you make a level-0 backup, perform a complete backup of the logical log, including the current logical-log file.

If you use ON-Archive, execute the following command to make a full-system, level-0 backup:

```
Onarchive> ARCHIVE/DBSPACESET=*
```

◆

For details about making backups, refer to your *Backup and Restore Guide* for ON-Bar information or to your *Archive and Backup Guide* for **ontape** or ON-Archive information.

UNIX

Unload the Data

You can unload your data from your source database server into ASCII or internal Informix format, which yields the fastest unload. You can use the UNLOAD statement, the High-Performance Loader (HPL), or **dbexport**. The HPL is the fastest of these methods. For information about these options to unload your data, refer to [Chapter 2, “Data Migration.”](#)

When unloading floats or small floats to a fixed-position file, if the internal precision of the float is greater than the fixed field can support, the number is truncated to fit the field. If you want more control over the formatted result of a float, use the SQL functions ROUND() or TRUNC() to define the output precision and to convert the float to a decimal.

Take OnLine Dynamic Server 7.2x Off-Line

Execute the following command to take OnLine Dynamic Server 7.2x to off-line mode:

```
onmode -ky
```

Bring your source database server off-line to ensure that all common files are inactive. Your source database server must be off-line because Dynamic Server with AD and XP Options uses the same files. You cannot install Dynamic Server with AD and XP Options if any of the files that it uses are active.

After you shut down your source database server, execute the following command to verify that it is off-line:

```
onstat -
```

Verify that you get the message `shared memory not initialized... for off-line mode`.



Important: Make a final backup for each source database server instance that you plan to upgrade.

UNIX

Change UNIX Kernel Parameters

You might need to change some of the kernel parameters for your operating system before you install Dynamic Server with AD and XP Options. To reconfigure the operating system, follow the directions in the machine notes file included in your distribution media for your operating system.

For information on the location of the machine notes file, refer to “[Documentation Notes, Release Notes, and Machine Notes](#)” on page 22 of the Introduction.



Important: Make sure you modify the kernel parameters on every node.

Install Dynamic Server AD/XP 8.21

To install Dynamic Server with AD and XP Options on UNIX, you must log in as user **root** or **informix**. To install Dynamic Server with AD and XP Options on Windows NT, you must be a member of the **Informix-Admin** group. Set the **INFORMIXDIR** environment variable to the directory where you plan to install the database server. Install it on the node that contains the connection coserver in the directory that was NFS mounted on all the other nodes. Install the entire distribution on a single node within a file system that is shared across all the nodes that are assigned to Dynamic Server with AD and XP Options.

Check that the file system can hold the entire Dynamic Server with AD and XP Options distribution of approximately 180 megabytes. Export this file system with write access as user **root** or **informix** over the NFS and mount it to the same mount point on every node.



Warning: If you install Dynamic Server with AD and XP Options in the same directory where OnLine Dynamic Server resides, the installation script overwrites the older files. If you want to preserve your OnLine Dynamic Server files, you must install Dynamic Server with AD and XP Options in a different directory on UNIX or on different computers on Windows NT.

Check the release notes for information about the proper operating-system patches, recommended shared-memory parameters, and configurations that are required for successful installation and operation of the database server.

The release notes are in the **\$INFORMIXDIR/release/en_us/0333** directory. ♦

UNIX

WIN NT

The release notes are in %**INFORMIXDIR**%\release\en_us\0333. ♦

Make any required changes that the release notes recommend on every node. For more information on how to install Dynamic Server with AD and XP Options, refer to your *Installation Guide* and your *Performance Guide*.

Set Environment Variables

After you install Dynamic Server with AD and XP Options and before you invoke DB-Access, set up the following environment variables on every node:

- **INFORMIXDIR**
- **INFORMIXSERVER**
- **ONCONFIG**
- **PATH**
- **INFORMIXSQLHOSTS**

The client application looks for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file. ♦

The client application looks for connectivity information in a key in the Windows NT registry named **HKEY_LOCAL_MACHINE\SOFTWARE\Informix\SQLHOSTS**. ♦

Set the environment variable **PATH** so that the local directory that contains the Informix commands is searched before the **INFORMIXDIR** directory.

The installation script installs Dynamic Server with AD and XP Options into the **INFORMIXDIR** directory specified for user **root** or **informix** on UNIX platforms or for a member of the **Informix-Admin** group on Windows NT platforms. The installation script does not bring Dynamic Server with AD and XP Options on-line.

UNIX

WIN NT

Copy Database Server Utilities to Each Node

To ensure rapid and proper initialization, create local copies of the following utilities on each node:

- **oninit**
- **onmode**
- **onstat**

When you finish copying the utilities to each node, log out as **root**.

Update the ONCONFIG Configuration File

You can customize your **ONCONFIG** configuration file and environment variables to take advantage of the new features that Dynamic Server with AD and XP Options introduced. After you observe the performance of Dynamic Server with AD and XP Options, you might want to make further adjustments.

For configuration information, refer to your *Administrator's Guide*. For information about environment variables, refer to the *Informix Guide to SQL: Reference*.

Update the sqlhosts File or Registry Key

The **sqlhosts** file or registry key contains a line for each connection type that the database server provides, for each coserver that makes up Dynamic Server with AD and XP Options, and for each 8.21 database server to which a client connects. An **sqlhosts** entry must exist for every coserver.

For details, see [“The sqlhosts File or Registry Key” on page 10-8](#).

Update the Backup and Restore Configuration Parameters

You might need to change the values of the backup and restore configuration parameters. [Figure 10-1 on page 10-6](#) lists these parameters.

For more information, see [“Configuration Parameters” on page 10-5](#).

Bring Dynamic Server AD/XP 8.21 On-Line

Execute the following command to bring Dynamic Server with AD and XP Options 8.21 on-line for the first time:

```
xctl -C oninit -iy
```

Review the message logs to verify that your **sysmaster** database and **sysmaster** catalog tables were created successfully.

Check your database server message log to verify that all coservers are up. To view the message log, run:

```
onstat -m
```

To verify that all your coservers are up and running, execute:

```
xctl onstat -
```

For information about messages in the message log, refer to your *Administrator's Guide*.

Important: If the message file indicates problems, solve the problems before you continue to the next step.



Use onutil to Create Cogroups and Dbslices

Create cogroups and dbslices of equal sizes across all coservers. When you create a dbslice, you specify the cogroup name so that Dynamic Server with AD and XP Options knows the coservers on which to create dbspaces. For example, you might create a dbslice from an accounting cogroup. The following example shows how to create a cogroup and a dbslice:

```
% onutil
1> CREATE COGROUP acctg_group
:
:
5> CREATE DBSLICE acctg_dbslc
6> FROM acctg_group ...
```

You do not need to specify the names explicitly for all the individual dbspaces that are associated with the partitioned tables. Dynamic Server with AD and XP Options generates the dbspace names for you.

Run **xctl** to verify that all your dbspaces were created correctly:

```
xctl onstat -d
```

Create a Database

Create a database instance on Dynamic Server with AD and XP Options and choose the optimal loading scheme from the data-loading options discussed in [“Loading and Unloading Data” on page 10-19](#). After you install Dynamic Server with AD and XP Options, you can create a database using DB-Access as your client application or issue SQL statements to connect to and disconnect from Dynamic Server with AD and XP Options.

You can start IECC from a Windows NT console to customize one or more database servers. For instructions on how to customize a database server, see your *Administrator's Guide*.

Load Data from External Tables

The following section describes how to load data into Dynamic Server with AD and XP Options external tables and convert it into internal Informix format. You can issue these SQL statements in the DB-Access utility or embed them in Informix ESQL/C.

Use a CREATE TABLE statement to create a table in your database to which you plan to load the data from OnLine Dynamic Server. For example:

```
CREATE TABLE account
  (account_id integer,
   account_bal integer,
   account_date date,
   account_name char(30)
  )
FRAGMENT BY HASH(account_date)
IN account_dbsp1,
   account_dbsp2,
   :
   account_dbspn;
```

Issue a series of SQL statements to create external tables for loading data. The CREATE EXTERNAL TABLE statement describes the location of the external file (on file or from pipe) and the format of external data. When you load with external tables, you can select FILE or PIPE for named pipes for your data source. You can load DELIMITED format data, FIXED format data or Informix internal format data (which is created when you unload data in this format). Loading with Informix internal format is fast because no conversion is required. Delimited or fixed-format data can be in ASCII or EBCDIC code sets.

The external table statement can also describe data-conversion information and specify a file for error and diagnostic information.

For example:

```
Create EXTERNAL TABLE extTableName
    SAME AS sourceTableName
    USING
    FORMAT "informix",
    DATAFILES
(
    "DISK: COGROUP_ALL:/data/file.%c",
    REJECTFILE "/pload/reject"
);
```

If the files you are loading were unloaded from a database in Informix internal format, specify an Informix format in the USING clause. The SAME AS clause creates the external table with the same definitions for the columns of the data files from the existing database.

Specify DATAFILES

The USING DATAFILES clause specifies the file type, coserver or cogroup, and the location of the file, and the file format definition and location for a reject file for error and diagnostic information. The USING DATAFILES clause takes double quoted strings separated by commas. The DATAFILES statement has three parts, separated by colons: the data type, the data location, and the absolute path to the file.

To move data between external tables and internal tables, issue data manipulation language (DML) SQL statements. Load data files into the database by issuing the INSERT AND SELECT statements. The INSERT and SELECT statements map the movement of the external data from or to the database table. The data in the data files is loaded from the external table into an internal table using the INSERT into SELECT statements. For example:

```
LOCK TABLE IN EXCLUSIVE MODE; # for deluxe mode

INSERT INTO account
(account_id, account_bal, account_date date, account_name)
SELECT *
FROM extTableName
USING
(FORMAT "informix",
DATAFILES
(
"DISK:3:/tmp/data.1"
"DISK:5:/tmp/data.1"
)
WHERE account_name [1] = "A";
```



Tip: Specify the format `informix` only for data files you unload from an Informix database. The `WHERE` clause can specify which data file rows to load.

Set Log File and Maximum Errors

You can set a log file that stores session information and statistics gathered about each load and unload when it completes. The log file also lists reject files. For information on reject files, see [“Check Rejection Files” on page 10-37](#). If the WITH APPEND keywords are used, the new log information is appended to an existing file; otherwise the file is truncated. To set this log file, issue the following statement:

```
SET PLOAD FILE TO filename
[WITH APPEND];
```



Tip: You can only have one file open at a time. If you issue multiple SET PLOAD FILE statements, the last one you issue is in effect.

To control the maximum number of errors allowed per coserver before a load is aborted, you can set the MAXERRORS option in the external table. Include the following statement in your CREATE EXTERNAL TABLE statement:

```
MAXERRORS maxNumErrPerCoserver
```

For more information on the CREATE EXTERNAL TABLE statement, refer to the *Informix Guide to SQL: Syntax*.

You can also use **onstat -g xmp** and **onstat -g dfm** to monitor the query segments and data flow. For more information, see your *Performance Guide*.

Run UPDATE STATISTICS and Build Indexes

Run UPDATE STATISTICS on all the tables you load into your database.

Build indexes on the tables in Dynamic Server with AD and XP Options that were indexed in OnLine Dynamic Server. You might want to wait before you build indexes to see if your queries run fast enough without them.

Verify the Integrity of the Data

After Dynamic Server with AD and XP Options finishes upgrading the system catalog tables, use **onutil** to verify that no data was corrupted in the migration process. The **onutil** commands are not SQL statements; neither DB-Access nor any other SQL utility or application supports them. Only user **informix** or **root** can use the CHECK and DISPLAY DATA clauses. The **onutil** CHECK options place a shared lock on tables when they check indexes. They also place shared locks on system catalog tables when they check them. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes, as [Figure 10-10](#) shows.

Figure 10-10
Commands for Verifying Data Integrity

Action	onutil Command
Check reserve pages	onutil CHECK RESERVED DISPLAY DATA
Check extents	onutil CHECK SPACE DISPLAY DATA
Check system catalog tables	onutil CHECK CATALOGS
Check data	onutil CHECK TABLE DATA DISPLAY DATA
Check indexes	onutil CHECK INDEX DISPLAY DATA

You might want to test and run your queries on Dynamic Server with AD and XP Options and make sure they produce the same results.

For more information about **onutil**, see your *Administrator's Guide*.

Make an Initial Backup of Dynamic Server AD/XP 8.21

Use your backup and restore tool (ON-Bar) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of your source database server.

Dynamic Server with AD and XP Options supports unloading to pipes, so you can unload directly to a tape device. You do not need to unload data first to disk.

You can use external tables to perform a backup. This process requires adequate disk space to stage your data for loading and unloading. To back up, unload a table to an external table and back up to tape. To restore, load data from your external tables.

For more information about ON-Bar, see your *Backup and Restore Guide*.



Important: Do not restore the backed up logical-log files from OnLine Dynamic Server for Dynamic Server with AD and XP Options.

Check Rejection Files

The REJECTFILE file specifies which rows were rejected during a load for each coserver. Each coserver must have a unique REJECTFILE name. Enter the REJECTFILE parameter in the USING clause of the EXTERNAL table definition as the example on [page 10-33](#) shows. For example:

```
REJECTFILE absolutePathName
```



Warning: Do not use shared files across coservers to ensure that the data in the reject files retains integrity.

Check the REJECTFILE file from your load command and see if any records were not loaded during the load. If necessary, manually update tables. You can also monitor message files using **onstat -g xmf**. For more information on how to monitor options, see your *Performance Guide*.

Recompile ESQL/C Programs

Informix ESQL/C enables programmers to embed SQL statements directly into a C program. Once your database server migration is complete, recompile your ESQL/C programs. Make sure to set the environment variables before you invoke an ESQL/C program.

When the ESQL/C program is recompiled, check the functionality of your ESQL/C programs to make sure they run as you expect them to. For more information, see your *INFORMIX-ESQL/C Programmer's Manual*.

Complete Migration

When you finish the level-0 backup, the migration process is complete and users can use Dynamic Server with AD and XP Options to access data safely.

Once you successfully migrate to Dynamic Server with AD and XP Options, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between OnLine Dynamic Server 7.2x and Dynamic Server with AD and XP Options. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on performance topics, refer to your *Performance Guide*.

Reverting to OnLine Dynamic Server 7.2x from Dynamic Server AD/XP 8.21

After you migrate to Dynamic Server with AD and XP Options from OnLine Dynamic Server 7.2x, you cannot revert back automatically. To revert, you need to unload the data into an external table from Dynamic Server with AD and XP Options and then load it back into OnLine Dynamic Server.

Unloading Data to External Tables

To unload data to an external file, issue SELECT and INTO EXTERNAL statements. You can implicitly or explicitly specify an external table. (If you are loading data, an explicit definition is mandatory.) The SELECT and INTO EXTERNAL statements create a default external table description for unloading data. Issue the following SQL statement:

```
SELECT *
FROM customer
WHERE customerNum > 100
AND lastName [1] = "A"
INTO EXTERNAL TABLE custExtII
USING
(
  FORMAT "informix",
  DATAFILES
  ( "DISK:1:/tmp/dat.out",
    "DISK:2:/tmp/dat.out",
    "DISK:3:/tmp/dat.out"
  ));
```

In this example, the external table is implicitly defined when the table is unloaded. If the external table already exists, you can use the following syntax:

```
INSERT INTO extTableName
SELECT *
FROM tableName
WHERE...;
```

For more information about external tables, see your *Administrator's Guide*.

Loading Data from External Tables

After the data is unloaded into external tables, you can use OnLine Dynamic Server load utilities to load the data. For load utility information, see [Chapter 2, "Data Migration."](#)

Version 7.x Database Server Migration

- Chapter 11** **Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0**
- Chapter 12** **Migrating Between a 7.x or 6.0 Database Server and a 5.x or 4.1 Database Server**
- Chapter 13** **Migrating Between INFORMIX-SE Database Servers and Converting C-ISAM Files**
- Chapter 14** **Migrating Between 7.x Database Server Options and Editions**



Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0

In This Chapter	11-5
Changes in Dynamic Server 7.3 and OnLine Dynamic	
Server 6.0 to 7.2	11-6
Environment Variable Changes in Dynamic Server 7.3	11-6
New Configuration Parameters in Dynamic Server 7.3	11-7
New SQL Reserved Words in Dynamic Server 7.3	11-7
New Features in Dynamic Server 7.3	11-8
New Functions	11-9
Upgrading to ON-Bar from ontape or ON-Archive	11-9
Multiple Residency and High Availability	11-9
Environment Variable Changes in OnLine Dynamic	
Server 7.2	11-10
New Environment Variables in OnLine Dynamic	
Server 7.2	11-11
Environment Variables for Backward Compatibility	11-12
Configuration Parameter Changes in OnLine Dynamic	
Server 7.2	11-12
New Configuration Parameters in OnLine Dynamic	
Server 7.22	11-13
Configuration Parameter Changes in OnLine Dynamic	
Server 7.22	11-13
New Configuration Parameters in OnLine Dynamic	
Server 7.2	11-13
Configuration Parameters That Online Dynamic	
Server 7.2 Dropped	11-13
In-Place ALTER TABLE New in OnLine Dynamic	
Server 7.24	11-14
Enhancements to sqlhosts File or Registry Key in OnLine	
Dynamic Server 7.23.	11-14

Language Support Changes in OnLine Dynamic	
Server 6.0 to 7.2	11-15
Environment Variable Changes in OnLine Dynamic	
Server 7.10.UD1	11-16
Configuration Parameter Changes in OnLine Dynamic	
Server 7.10.UD1	11-16
New Configuration Parameters in OnLine Dynamic	
Server 7.10.UD1.	11-16
Configuration Parameters That Were Dropped.	11-17
Configuration Parameters That Were Moved	11-17
Changes to BlobSpace Requirements in OnLine Dynamic	
Server 7.10.UD1	11-17
Environment Variable Changes in OnLine Dynamic	
Server 7.1	11-18
Configuration Parameter Changes in OnLine Dynamic	
Server 7.1	11-18
ON-Archive Changes in OnLine Dynamic Server 7.1	11-19
Environment Variable Changes in OnLine Dynamic	
Server 6.0	11-19
Preparing for Migration	11-19
Install the Latest Maintenance Release for the Current	
Version of OnLine Dynamic Server.	11-21
Verify and Install a Validated Storage Manager	11-21
Check Available Space and System Requirements	11-22
Save Copies of the Current Configuration Files	11-22
Shut Down the Source Database Server	11-23
Verify the Integrity of the Data	11-24
Back Up the Source Database Server Files.	11-25
Begin Migration.	11-25
Upgrading to Dynamic Server 7.3	11-26
Save an Output File of SQL Statements for Access Paths.	11-27
Bring the Source Database Server Off-Line	11-27
Change UNIX Kernel Parameters.	11-27
Install the Target Database Server	11-27
Install Informix Enterprise Command Center	11-28
Customize the Target Database Server Environment	11-29
Update the ONCONFIG Configuration Parameters	11-29

Update the Configuration Files	11-30
Prepare the Target Database Server to Use Enterprise Replication	11-30
Install and Configure SNMP	11-31
Bring the Target Database Server On-Line	11-32
Update Statistics	11-33
Verify the Integrity of the Data	11-33
Check Space Requirements for BYTE and TEXT Data	11-33
Make an Initial Backup of the Target Database Server	11-33
Verify the Access Path of the SQL Statements	11-34
Complete Migration.	11-34
Change Database Server Definitions	11-35
Migrating to a GLS Locale	11-35
Reverting to an Earlier Version of OnLine Dynamic Server.	11-36
Remove Unsupported SQL Features	11-38
Remove Fragmented Tables (Version 6.0 Only)	11-38
Save System Catalog Information	11-38
Stop Enterprise Replication or Workgroup Replication (Version 7.22 or Later).	11-39
Uninstall the Database Server and Informix Enterprise Command Center	11-40
Modify In-Place ALTER TABLE.	11-40
Close All Transactions and Put the Source Database Server in Quiescent Mode	11-41
Verify the Integrity of the Data	11-42
Back Up the Source Database Server	11-42
Run the Reversion Utility	11-43
Remove GLS Features	11-44
Modify Configuration Parameters	11-44
Configuration Changes for OnLine Dynamic Server 7.22	11-45
Configuration Changes for OnLine Dynamic Server 7.12 and 7.22	11-45
Configuration Changes for OnLine Dynamic Server 7.10.UD1 Through 7.14	11-45

Configuration Changes for OnLine Dynamic Server 7.10.UC1.	11-46
Configuration Changes for OnLine Dynamic Server 6.0	11-47
Reset Environment Variables	11-48
Environment Variable Changes for OnLine Dynamic Server 7.10.UD1 Through 7.14	11-48
Environment Variable Changes for OnLine Dynamic Server 7.10.UC	11-49
Environment Variable Changes for OnLine Dynamic Server 6.0	11-49
Modify the sqlhosts File	11-50
Reinstall the Earlier Version of the Database Server	11-50
Bring the Target Database Server On-Line.	11-50
Rename ON-Archive Files (Version 6.0 Only)	11-51
Restore System Catalog Information (Version 6.0 Only)	11-51
Verify the Integrity of the Data	11-51
Back Up the Target Database Server Files	11-52
Return the Target Database Server to On-Line Mode	11-52
Complete Reversion	11-52
Reverting to OnLine Dynamic Server 7.10.UCx from OnLine Dynamic Server 7.10.UDI Through 7.14	11-53
Verify the Integrity of the Data	11-53
Back Up the Target Database Server.	11-54
Complete Reversion	11-54

In This Chapter

This chapter describes the procedures to migrate between Dynamic Server, Version 7.31, and Dynamic Server, Version 7.30, or between Dynamic Server, Version 7.3x, and OnLine Dynamic Server, Version 6.0 to Version 7.24.

You can use the same migration procedures to upgrade to a 7.1x or 7.2x database server from an earlier version or to revert to the earlier version, if you note the version-specific features and any necessary additional steps. When you upgrade to a newer version of a database server, you might want to upgrade database applications to take advantage of new features. The release notes files that accompany the software describe new features.

This chapter covers the following topics:

- Changes introduced by versions 6.0 to 7.3 of the database server
- Preparing for migration
- Upgrading your database server to Dynamic Server 7.3
- Migrating to a GLS locale
- Reverting your database server to an earlier version

You can upgrade from or revert to the following versions:

- OnLine Dynamic Server, Version 6.0, Version 7.10.UCx, Version 7.10 UD1, (UNIX only) through Version 7.14, and Version 7.20 through Version 7.24
- OnLine Dynamic Server, Version 7.12 ♦

Tip: For information on how to upgrade from a database server prior to Version 6.0, refer to [Chapter 12, “Migrating Between a 7.x or 6.0 Database Server and a 5.x or 4.1 Database Server.”](#) For information on how to move the database server data between UNIX and Windows NT, refer to [Chapter 15, “Moving a Database Server to a Different Operating System.”](#)

WIN NT



Changes in Dynamic Server 7.3 and OnLine Dynamic Server 6.0 to 7.2

This section describes changes to environment variables and configuration parameters and discusses database server functionality that affects migration. Different versions of Informix database servers introduced the changes, which this section describes in descending order, starting with the changes to Dynamic Server 7.3. For each database server version, this section discusses the environment variables and configuration parameters and lists the new features or changes to existing features for that version.

Environment Variable Changes in Dynamic Server 7.3

Dynamic Server 7.3 introduced several new environment variables and maintains several otherwise obsolete environment variables for backward compatibility. The *Informix Guide to SQL: Reference* describes the following new environment variables:

- **INFORMIXKEYTAB**
- **IFX_DIRECTIVES**

Dynamic Server 7.3 includes the Informix Storage Manager (ISM), a backup and recovery media management utility, with new environment variables. For more information, refer to the *Informix Storage Manager Administrator's Guide*.

UNIX

New Configuration Parameters in Dynamic Server 7.3

Dynamic Server 7.3 introduced the following new configuration parameters. The *Guide to Informix Enterprise Replication* describes configuration parameters that start with CDR. The *Informix Storage Manager Administrator's Guide* describes ISM parameters. Your *Performance Guide* describes optimizer configuration parameters. Your *Administrator's Guide* describes other parameters.

- CDR_DSLOCKWAIT
- CDR_EVALTHREADS
- CDR_LOGBUFFERS
- CDR_QUEUEMEM
- DIRECTIVES (Optimizer directives)
- ISM_DATA_POOL
- ISM_LOG_POOL
- OPT_GOAL
- RESTARTABLE_RESTORE (Defaults to Off)
- SYSALARMPROGRAM
- TBLSPACE_STATS ♦
- LOG_BACKUP_MODE
- OPTICAL_LIB_PATH (For new optical functionality)
- STAGEBLOB (New for Windows NT) ♦

WIN NT

New SQL Reserved Words in Dynamic Server 7.3

Dynamic Server 7.3 supports new SQL keywords that are reserved words and might affect migration of your applications. Although you can use almost any word as an SQL identifier, syntactic ambiguities can occur. An ambiguous statement might not produce the results you want.

This section lists the new SQL reserved words. For more information about SQL reserved words, see the *Informix Guide to SQL: Syntax*.

The following SQL keywords are new in Dynamic Server 7.31:

- INNER
- JOIN
- LEFT
- LOCKS
- RETAIN

The following SQL keywords are new in Dynamic Server 7.30:

- ALL_ROWS
- CASE
- CRCOLS
- DECODE
- FIRST_ROWS
- MEMORY_RESIDENT
- NON_RESIDENT
- NVL
- REPLICATION
- SUBSTR
- SUBSTRING

New Features in Dynamic Server 7.3

Dynamic Server 7.3 introduced several features that affect migration. These features include new built-in functions to the database server, changes to backup and restore mechanisms, and multiple residency options on Windows NT. For Dynamic Server 7.3, the Informix Enterprise Command Center (IECC) interface is available on UNIX and Windows NT.

New Functions

The following new built-in functions are available in Dynamic Server 7.3 to facilitate migration to Informix database servers:

- | | | |
|------------------|------------------|--------------------|
| ■ DECODE | ■ TO_CHAR | ■ SUBSTR |
| ■ NVL | ■ LPAD | ■ SUBSTRING |
| ■ TO_DATE | ■ RPAD | ■ REPLACE |

The names of these functions are reserved words in the database server. For more information on these functions, refer to the *Informix Guide to SQL: Syntax*.

Upgrading to ON-Bar from ontape or ON-Archive

Dynamic Server 7.3 includes ON-Bar, **ontape**, and ON-Archive for backup and recovery. If you migrate from a previous version of the database server, you might want to upgrade from **ontape** or ON-Archive to ON-Bar. For more information on ON-Bar, refer to your *Backup and Restore Guide*.

In Version 7.3, ON-Bar uses a new ISM to read data from or to write data to tape or disk. You can administer ISM through the ISM Administrator, which you can start from IECC. For more information on ISM, refer to the *Informix Storage Manager Administrator's Guide*.

Important: To restore data, always use the same backup and restore method you used to create your backup.



WIN NT

Multiple Residency and High Availability

As of Dynamic Server 7.30, *multiple residency* is available on Windows NT. Multiple residency allows multiple database servers and their associated shared memory and disk structures to coexist on a single computer. Each instance of the database server has its own **INFORMIXSERVER**, **INFORMIXDIR**, and **ONCONFIG** values.



In Dynamic Server 7.3 on Windows NT, multiple residency enables the database server to be installed as a cluster-aware application, which in turn can facilitate high availability. A cluster-aware application is an application that is registered with the resource manager and provides dynamically linked libraries (resource DLLs).

Tip: Prior database server versions cannot co-exist with Dynamic Server 7.3 because multiple residency is available only as of Version 7.30. If a prior version is present, you must uninstall and re-install the new version to migrate on Windows NT. The installation process uses configuration files from the previous version. For more information on how to install Version 7.3 on Windows NT, see your *Installation Guide*.

High availability provides redundant components in a cluster of two or more nodes at an operating-system level. Whenever a failure occurs on one node (one Windows NT computer), the cluster manager restarts the failed application on the surviving node. The cluster manager and resource manager detect failures and communicate to other subsystems and applications.



Tip: When you upgrade database server versions, you might need to reinstall backups of the target database server to ensure that an entire cluster is upgraded.

For more information on multiple residency and high availability, refer to your *Administrator's Guide*.

Environment Variable Changes in OnLine Dynamic Server 7.2

OnLine Dynamic Server 7.2 introduced several new environment variables and maintains several otherwise obsolete environment variables for backward compatibility.

New Environment Variables in OnLine Dynamic Server 7.2

Figure 11-1 shows new environment variables for Version 7.2. Review the descriptions of these environment variables to determine whether you need to set them. The **Reference** column in Figure 11-1 indicates the manuals that provide information about these environment variables. Figure 11-1 uses the following abbreviations for Informix manuals:

- HPL: *Guide to the High-Performance Loader* (The High-Performance Loader (HPL) is available on Windows NT as of Version 7.30.)
- GLS: *Informix Guide to GLS Functionality*
- ESQ/C: *Informix ESQ/C Programmer's Manual*
- REF: *Informix Guide to SQL: Reference*

Figure 11-1
Environment Variables Introduced in Version 7.2x

Environment Variable	Variable Affects	Reference
CC8BITLEVEL	ESQ/C only	GLS
CLIENT_LOCALE	Client applications only	GLS
DBCENTURY	SQL APIs only	REF, GLS
DBONPLOAD	High-Performance Loader only	REF, HPL
DB_LOCALE	Database locale	GLS
ESQLMF	ESQ/C compilation	GLS
GLS8BITFSYS	8-bit clean	GLS
GL_DATE	Date format	GLS
GL_DATETIME	Time format	GLS
IFX_AUTOFREE	ESQ/C compilation	ESQ/C
NODEFDAC	Default privileges	REF
ONPLOAD	High-Performance Loader	REF, HPL
OPTOFC	ESQ/C compilation	ESQ/C

(1 of 2)



NLS

Environment Variable	Variable Affects	Reference
PLCONFIG	High-Performance Loader	REF, HPL
SERVER_LOCALE	Database server locale	GLS
THREADLIB	ESQL/C only	REF, ESQL/C

(2 of 2)

Tip: The IFX_AUTOFREE environment variable introduced in Version 7.22 is discontinued in future releases. If you want to take advantage of this feature, use SET AUTOFREE syntax in your application.

Environment Variables for Backward Compatibility

OnLine Dynamic Server 7.2 supports the environment variables in the following list for backward compatibility with earlier Informix products. If you do not have databases and applications from pre-7.2 versions, you would not use these environment variables. Version 7.1 of the *Informix Guide to SQL: Reference* describes these environment variables. The *Informix Guide to GLS Functionality* describes the interaction of these environment variables with variables that control GLS. The following environment variables maintain backward compatibility:

- COLLCSHAR
- DBAPICODE
- DBNLS
- LANG
- LC_COLLATE
- LC_CTYPE
- LC_MONETARY
- LC_NUMERIC
- LC_TIME

Configuration Parameter Changes in OnLine Dynamic Server 7.2

OnLine Dynamic Server 7.2 includes new configuration parameters that might affect your installation. You might also need to adjust the values of existing parameters. These configuration parameters are described in your *Administrator's Guide*.

New Configuration Parameters in OnLine Dynamic Server 7.22

Enterprise Replication (and Workgroup Replication for OnLine Workgroup Server, Version 7.22, and OnLine Dynamic Server, Version 7.22, on Windows NT only) use the following new configuration parameters:

- | | |
|-------------------|------------------|
| ■ CDR_LOGBUFFERS | ■ CDR_NIFUSEHELP |
| ■ CDR_EVALTHREADS | ■ CDR_NIFMEMS |
| ■ CDR_DSLOCKWAIT | ■ CDR_NIFQUEUES |
| ■ CDR_QUEUEMEM | |

For more information on these configuration parameters, see the *Guide to Informix Enterprise Replication*.

Configuration Parameter Changes in OnLine Dynamic Server 7.22

The default value of ALARMPROGRAM has changed.

New Configuration Parameters in OnLine Dynamic Server 7.2

Version 7.20 or later uses the following new configuration parameters:

- BUFFERS (new definition)
- HETERO_COMMIT
- MAX_PDQPRIORITY (new definition)

Configuration Parameters That Online Dynamic Server 7.2 Dropped

Version 7.20 or later does not use the following configuration parameters:

- BUFFSIZE
- PDQPRIORITY (the default value is always zero)

If you do not set PDQPRIORITY through the environment variable or SQL statement, PDQPRIORITY is turned off for queries (PDQPRIORITY = 0). For information on where to set environment variables, refer to the *Informix Guide to SQL: Reference*.

In-Place ALTER TABLE New in OnLine Dynamic Server 7.24

Version 7.24 of OnLine Dynamic Server introduced an In-Place option to the ALTER TABLE statements that was not available in previous versions. With the In-Place modifier, you can change a table without creating a duplicate copy of the table. For the procedure to revert to a version of the database server earlier than Version 7.24, see [“Modify In-Place ALTER TABLE” on page 11-40](#). For more information on In-Place ALTER TABLE statements, see your *Performance Guide*.

Enhancements to sqlhosts File or Registry Key in OnLine Dynamic Server 7.23

Version 7.23 of OnLine Dynamic Server introduced formatting changes and new syntax options in the **sqlhosts** file or registry key. The fifth field, the **options** field, was modified to accommodate new syntax options.

The following list is a review of the **sqlhosts** or registry fields:

FIELD 1	FIELD 2	FIELD 3	FIELD 4	FIELD 5
dbservername	nettype	hostname	servicename	options

The **options** field can contain columns separated by a comma or white space that represents the end of the column. Client and database server applications check each column to determine whether the option is supported in the database server release.

In earlier versions, the **options** field could contain only a single character. Version 7.23 and later versions of the database server support longer names. Earlier versions, such as Version 7.10, cannot process the longer syntax.



***Tip:** If you maintain more than one version of the database server on UNIX, use separate **sqlhosts** file entries for each version. Alternatively, you can use separate entries with an alias to the appropriate database server. You cannot maintain two versions of the database server on Windows NT.*

[Figure 11-2](#) lists the OnLine Dynamic Server 7.23 and later **sqlhosts** file or registry key components.

Figure 11-2
sqlhosts File or Registry Fields

dbservername	nettype	hostname	servicename	options
dbservername.server1	olsoctcp	clipper	14000	k=1,r=1 b=2000

In the sample **sqlhosts** file, the **options** field contains three options in three columns.

Column	Option
Column 1	k=1
Column 2	r=1
Column 3	b=2000



Important: Informix recommends that you use field 5, **options**, in Version 7.23 and later, for the following options only: **b**, **k**, **r**, **s**. If you do not want any of these options but do want other options, use **k=1** in column 5, which is the default. Place other options in subsequent columns.

For more information on the components of the **sqlhosts** file or registry key or on how to define two **sqlhosts** files, see your *Administrator's Guide*.

Language Support Changes in OnLine Dynamic Server 6.0 to 7.2

NLS

Version 6.0 of OnLine Dynamic Server introduced Native Language Support (NLS). NLS supports single-byte locales but not multibyte locales. ♦

WIN NT

Version 7.12 of OnLine Dynamic Server for Windows NT supports NLS. ♦

GLS

Informix Version 7.2x products use Global Language Support (GLS). GLS enables Version 7.2x and later versions of the database server to handle different languages, cultural conventions, and code sets for Asian, European, Latin American, and Middle Eastern countries. [Chapter 17, “Understanding Language Support in Informix Database Servers,”](#) and [Chapter 18, “Converting to GLS,”](#) discuss the migration implications of GLS. The *Informix Guide to GLS Functionality* provides a full description of GLS. ♦

UNIX

Environment Variable Changes in OnLine Dynamic Server 7.10.UD1

Version 7.10.UD1 of OnLine Dynamic Server introduced the following new environment variables:

- INFORMIXOPCACHE
- INFORMIXSQLHOSTS
- NODEFDAC
- OPTCOMPIND (new definition)
- PSORT_NPROCS (new definition)

Configuration Parameter Changes in OnLine Dynamic Server 7.10.UD1

Version 7.10.UD1 of OnLine Dynamic Server introduced several new configuration parameters and dropped several others.

New Configuration Parameters in OnLine Dynamic Server 7.10.UD1

Version 7.10.UD1 of OnLine Dynamic Server introduced the following configuration parameters:

- LBUPRESERVE
- ONDBSPACEDOWN
- OPCACHEMAX
- OPTCOMPIND (new default value)

Configuration Parameters That Were Dropped

Version 7.10.UD1 of OnLine Dynamic Server dropped the following configuration parameters. OnLine Dynamic Server allocates resources dynamically for the structures that these parameters controlled in previous releases.

- | | |
|---------------|----------------|
| ■ CHUNKS | ■ TBLSPACES |
| ■ DBSPACES | ■ TRANSACTIONS |
| ■ USERTHREADS | |

You might need to reset the value of the LOCKS configuration parameter because it previously depended on the value of TRANSACTIONS.

Configuration Parameters That Were Moved

Version 7.10.UD1 of OnLine Dynamic Server moved the following configuration parameters from the ONCONFIG file:

- | | |
|-----------|-----------|
| ■ ADTPATH | ■ ADTERR |
| ■ ADTSIZE | ■ ADTMODE |

UNIX

On UNIX, these configuration parameters are in the audit configuration file (`$INFORMIXDIR/aaodir/adtcfg.std`). ♦

WIN NT

On Windows NT, these configuration parameters are in the audit configuration file (`%INFORMIXDIR%\aaodir\adtcfg`). ♦

Changes to BlobSpace Requirements in OnLine Dynamic Server 7.10.UD1

Versions of OnLine Dynamic Server before 7.10.UD1 marked a partition blob page as full if the page was more than one-third full. Version 7.10.UD1 uses a threshold of one-half the page size.

In cases where partition blobs have a random size, both schemas use about the same amount of disk space. However, in certain situations the required disk space changes. If you have many partition blobs that are just larger than one-third of a page but less than one-half of a page, the new schema reduces your space requirements by a factor of two. On the other hand, if you have partition blobs that are just less than one-third of a page and others that are just less than two-thirds of a page, you might see an increase in disk requirements of about 33 percent.

Environment Variable Changes in OnLine Dynamic Server 7.1

Version 7.1 of OnLine Dynamic Server introduced the following environment variables:

- DELIMIDENT
- FET_BUF_SIZE

Configuration Parameter Changes in OnLine Dynamic Server 7.1

Version 7.1 of OnLine Dynamic Server introduced the following configuration parameters:

- | | |
|------------------|-------------------|
| ■ ALARMPROGRAM | ■ DS_TOTAL_MEMORY |
| ■ DATASKIP | ■ MAX_PDQPRIORITY |
| ■ DS_MAX_QUERIES | ■ OPTCOMPIND |
| ■ DS_MAX_SCANS | |

In addition, the PDQPRIORITY configuration parameter was introduced in Version 7.1 but dropped in Version 7.20.

In Version 7.1, the default values for the configuration parameters LTXHWM and LTXEHWL changed from 80 and 90 to 50 and 60, respectively. OnLine Dynamic Server initialization provides a warning if your **ONCONFIG** file contains values for these parameters greater than 50 and 60.

UNIX

ON-Archive Changes in OnLine Dynamic Server 7.1

The names of the ON-Archive error message and help files changed between Version 6.0 and Version 7.10. If you use a **config.arc** file from Version 6.0, you must change the filenames in **\$INFORMIXDIR/etc/config.arc**. If you use the default **Config.arc** file (with initial uppercase letter) that is installed with Version 7.1, you do not need to make any changes. For more information, refer to your *Archive and Backup Guide*.

Environment Variable Changes in OnLine Dynamic Server 6.0

Version 6.0 of OnLine Dynamic Server introduced name changes for environment variables and utilities. Environment variable names that began with **TB** in earlier versions begin with **ON** in OnLine Dynamic Server 6.0 or later. For instance, the **TBCONFIG** environment variable was replaced by the **ONCONFIG** environment variable.

Utility names that began with **tb** in earlier versions begin with **on** in OnLine Dynamic Server 6.0 and later. For instance, **tbcheck** was replaced by **oncheck**. For a complete list of utilities, refer to your *Administrator's Guide*.

If you have not already changed the names of these environment variables and utilities in your shell scripts and in the login files of your users, make the changes now.

Preparing for Migration

To prepare for migration between Dynamic Server 7.3 and OnLine Dynamic Server, you need to understand the Informix guidelines for migrating between database servers. You also need to know about any new features that might affect migration, which [“Changes in Dynamic Server 7.3 and OnLine Dynamic Server 6.0 to 7.2”](#) on page 11-6 describes.

When you migrate from one version of the database server to another, Informix recommends that you consider the following guidelines:

- Review the release notes for the version of your database server for information about new features, installation, and fixes to problems. Modify applications as needed.

The release notes are in one of the following directories:

- ❑ **\$INFORMIXDIR/release/en_us/0333**. ♦
- ❑ **%INFORMIXDIR%\release\en_us\0333** (as of Version 7.20)

Release Notes appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar. ♦

UNIX

WIN NT

- Retain both versions of the Informix product software on disk, if possible (if you have enough disk resources). On Windows NT, you can retain multiple versions of Informix products as of Dynamic Server 7.30.
- Check the documentation notes for information about features not covered in the manuals.
- Retain the installation media from both versions of the Informix product software.
- Make a level-0 backup of the database server before and after migration.
- Verify storage-manager validation for the target database server.

For details, see [“Verify and Install a Validated Storage Manager” on page 11-21](#).

For additional installation information and guidelines, refer to your *Installation Guide* and your *Getting Started* manual.

Before you migrate your database server, complete the following steps, which the next sections describe:

1. Install the latest maintenance release for the current version.
2. For Dynamic Server 7.3x, verify and, if necessary, install a storage manager.
3. Check available space and system requirements.
4. Save copies of the current configuration files.
5. Shutdown your database server.
6. Verify the integrity of the data.
7. Back up your database server files.

Important: Repeat steps 4 through 7 for each instance of your database server that you are migrating.



Install the Latest Maintenance Release for the Current Version of OnLine Dynamic Server

Informix recommends that you install the latest maintenance release for your current database server version before you migrate to a new version, especially if you use Version 5.0x or Version 6.0.

For example, to migrate to Dynamic Server 7.3 from OnLine 5.03 on UNIX, first install the latest maintenance release for OnLine and then migrate to Dynamic Server 7.3. Many minor changes to the 5.x versions are also in the 7.x versions.

For additional information, refer to the installation guide for your database server and the chapters on installation and configuration in your *Administrator's Guide*.

Verify and Install a Validated Storage Manager

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the **ism_catalog** utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Check Available Space and System Requirements

The database server requires 1100 free pages of logical-log space (around 2000 kilobytes) to build the **sysmaster** database. The Informix Dynamic Server 7.3 media is more than twice as large as previous versions because of extra GLS files.

WIN NT

Before you install your database server and IECC, verify that your system meets the minimum space and hardware requirements. Dynamic Server 7.3 requires 15 to 20 percent more space than the previous version because it includes administration tools, such as IECC and the Relational Object Manager.

The database server runs on Windows NT 4.0 with Service Pack 3 on an NTFS drive. The administration tools run on Windows NT 3.51, Windows NT 4.0, or Windows 95 on either a FAT drive or an NTFS drive. ♦

For information on the system requirements, refer to the **read_ods.txt** file in Answers OnLine, Version 1.7, or earlier.

Save Copies of the Current Configuration Files

Save copies of the configuration files for each instance of your database server. Keep the copies available to use later. Save the configuration files that [Figure 11-3](#) lists, if they exist.

Figure 11-3
ODS or IDS Configuration Files

UNIX	Windows NT
\$INFORMIXDIR/etc/\$ONCONFIG	%INFORMIXDIR%\etc\ONCONFIG
\$INFORMIXDIR/etc/onconfig	%INFORMIXDIR%\etc\onconfig
\$INFORMIXDIR/etc/onconfig.std	%INFORMIXDIR%\etc\onconfig.std
\$INFORMIXDIR/etc/sm_versions	%INFORMIXDIR%\etc\sm_versions
\$INFORMIXDIR/aaodir/adtcfg *	%INFORMIXDIR%\aaodir\adtcfg.*
\$INFORMIXDIR/dbssodir/adtmasks *	%INFORMIXDIR%\dbssodir\adtmasks.*
\$INFORMIXDIR/etc/sqlhosts	
\$INFORMIXDIR/etc/termcap	
\$INFORMIXSQLHOSTS	

UNIX

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

UNIX: `$INFORMIXDIR/etc/ixbar.<servername>`

Windows NT: `$INFORMIXDIR\etc\ixbar.<servername>`

If you use ON-Archive to back up and restore your database server and the logical logs, you must also copy and save the configuration files in the following list:

- `$INFORMIXDIR/etc/$ARC_CONFIG`
- `$INFORMIXDIR/etc/config.arc`
- `$INFORMIXDIR/etc/oper_deflt.arc`
- `$INFORMIXDIR/etc/tctermcap` ♦

If you use ON-Bar to back up and restore your database server and the logical logs, you must also copy and save the configuration files in `$INFORMIXDIR/etc/ixbar*`.

Shut Down the Source Database Server

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX. On Windows NT, you must be a member of the **Informix-Admin** group.
3. Execute the following command to take the database server to quiescent mode:

```
onmode -sy
```

4. Wait until your database server is in quiescent mode.

To verify the mode of your database server, execute the **onstat** - command. The first line of the **onstat** output contains the status of your database server. [Figure 11-4](#) shows that the database server is in quiescent mode.

```
INFORMIX-OnLine Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx -- xxxx  
  
OnLine Dynamic Server is in quiescent mode.  
Kbytes
```

Figure 11-4
Example of onstat
Status Line

5. Execute the following command to force a new logical log:

```
onmode -l
```

6. Execute the following command to force a checkpoint:

```
onmode -c
```

7. Execute the following command to shut down the database server:

```
onmode -yuk
```



Tip: Monitor your log activity to verify that all commands were executed properly and to check for inconsistencies prior to migration.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

Figure 11-5 lists the **oncheck** commands that verify data integrity. Informix recommends that at a minimum you run the **-cr** and **-ce** options.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 11-5
Commands for
Verifying the Data
Integrity

For information about **oncheck**, refer to your *Administrator's Guide*.

Back Up the Source Database Server Files

Use your preferred backup method (ON-Bar, **ontape**, or ON-Archive) to make a complete (level-0) backup of each database server that you plan to migrate.

The Windows NT environment does not support ON-Archive. ♦

For more information about how to use ON-Bar, **ontape**, or ON-Archive to back up your database server, refer to your *Backup and Restore Guide*.

Warning: Backups that you make under older versions of your database server are not compatible with the newer version. Do not try to restore these backups to the newer version.

Begin Migration

Now you are ready to migrate your database server. For information on how to upgrade, see [“Change Database Server Definitions” on page 11-35](#). For information on how to revert, see [“Reverting to an Earlier Version of OnLine Dynamic Server” on page 11-36](#).

WIN NT



Upgrading to Dynamic Server 7.3

This section describes how to upgrade to Dynamic Server 7.3 from OnLine Dynamic Server 6.0 or later. You should have completed the preparatory steps, described in [“Preparing for Migration” on page 11-19](#).

When you upgrade, you can install and test Dynamic Server 7.3 with the same database server name, configuration files, environment variables, and **sqlhosts** information or registry information that you used for the earlier version. After you install Dynamic Server 7.3 and verify that it works, you might want to modify the configuration files and environment variables to take advantage of new features.

When you upgrade your database server, complete the following steps, which are described in more detail in the sections that follow:

1. Save an output file of SQL statements for access paths.
2. Bring the source database server off-line.
3. Reconfigure the operating system, if necessary. ♦
4. Install the target version of your database server.
5. Install IECC.
6. Customize the database server environment.
7. Update the ONCONFIG configuration file.
8. Update the ON-Archive configuration files. ♦
9. Configure the database server for Enterprise Replication, if desired.
10. Install and configure **onsnmp**. ♦
11. Bring the target database server on-line.
12. Update statistics.
13. Verify the integrity of the data.
14. Check space requirements for BYTE and TEXT data.
15. Make an initial backup of your database server.
16. Verify the access paths of your SQL statements.
17. Complete migration.
18. Change database server definitions.

UNIX

UNIX

WIN NT

Save an Output File of SQL Statements for Access Paths

Save a file of output from any SET EXPLAIN statements. Later you can use this file to verify that access paths of your SQL statements do not change when you migrate to your target database server. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

WIN NT

For Windows NT, the SET EXPLAIN output filename is %INFORMIXDIR%\sqlexpln\username.out. ♦

Bring the Source Database Server Off-Line

Shut down your database server to ensure that all common files are inactive.

The database server must be off-line because the older and the newer versions share common files. You cannot install the database server if any of the common files are active.

WIN NT

The installation program automatically shuts down the old database server and starts the new database server. ♦

UNIX

Change UNIX Kernel Parameters

You might need to change some of the kernel parameters for your UNIX operating system before you install Dynamic Server 7.3. To reconfigure the operating system, follow the directions in the machine notes file included on your Dynamic Server 7.3 distribution media and the kernel-configuration instructions for your operating system.

Install the Target Database Server

On UNIX, you must be logged in as user **root**. On Windows NT, you must be a member of the **Administration** group to install your database server. Set the **INFORMIXDIR** environment variable to the directory where you plan to install your database server.

UNIX

The installation script installs your database server into the **INFORMIXDIR** directory specified for user **root**. The installation script does not bring the database server on-line. ♦

WIN NT



The setup program installs and brings up the database server on Windows NT. Follow the directions in your *Installation Guide* to install your database server. ♦

Warning: *If you install your database server in the same directory where the earlier version of the database server resides, the newer version overwrites the older files. If you want to preserve the files for the earlier version, you must install the newer database server in a different directory.*

Before you overwrite the older version, you must take the following precautions:

- If you do not have the original media for the older version, back up the **INFORMIXDIR** directory before you install your target database server.
- Copy the configuration file(s) in **INFORMIXDIR** in the **etc** directory to another location on the file system.

UNIX

When you finish the installation and system reconfiguration, exit as user **root** and log on as user **informix**. ♦

WIN NT

In Windows NT, you need to run the Installation wizard twice. First, upgrade the database server and then install the administration tools.

The Installation wizard replaces the files but does not reconfigure the database server. If a previous version of the database server is on the computer, the **Upgrade** page appears when you install the new product.

The installation program automatically verifies and brings down your source database server, copies the new files, and preserves the database configuration information. The installation program starts your target database server with the same configuration and shared-server computer. ♦

Install Informix Enterprise Command Center

After you install the database server, install the administration tool, IECC.

After you install the database server or IECC or both, IECC is available in your Windows NT environment. ♦

For information on how to install IECC, refer to the *Informix Enterprise Command Center Installation Guide*. For information about how to use IECC, refer to the *Informix Enterprise Command Center User Guide*.

Customize the Target Database Server Environment

After you install your database server, ensure that the following environment variables are set to the correct values:

- INFORMIXSERVER
- ONCONFIG
- PATH
- INFORMIXSQLHOSTS (if used)



WIN NT

Important: On UNIX, the client application looks for the **sqlhosts** file in the **etc** directory in the **INFORMIXDIR** directory. On Windows NT, **INFORMIXSQLHOSTS** points to the computer that contains the **sqlhosts** registry information. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

On Windows NT, the installation program sets the configuration parameters and environment variables for you. However, you can customize the configuration parameters in the %ONCONFIG% file and environment variables for your database server. In Windows NT, use **Setnet32** to customize the environment variables on the client computer. ♦

For information about environment variables, refer to the *Informix Guide to SQL: Reference*.

UNIX

Update the ONCONFIG Configuration Parameters

You can customize your ONCONFIG configuration file and environment variables to take advantage of the new features that Dynamic Server 7.3 introduced.

For example, you can add and adjust new ISM configuration parameters. After you observe the performance of your database server, you might want to make further adjustments.



Important: Use the same values for the target database server for **ROOTOFFSET**, **ROOTSIZE**, and **ROOTPATH** that you used for the source database server. Also, use the same values for size and number of physical logs, logical logs, and for mirroring (if available).

For information on how to configure, refer to your *Administrator's Guide*. For information about how to tune the configuration parameters, refer to your *Performance Guide*.

Update the Configuration Files

During the installation procedure, the install script checks the **etc** directory in the **INFORMIXDIR** directory for files named **config.arc**, **oper_deflt.arc**, **termcap**, **logevent.sh**, **sessalrm**, and **permalrm**. If these files do not exist, the install script provides them. If the files do exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc**, **Oper_deflt.arc**, **Termcap**, **Logevent.sh**, **Sessalrm**, and **Permalrm** (note the initial uppercase letters).

Compare your current versions of the files with the new versions and determine whether new or changed configuration parameters or qualifiers exist.

Prepare the Target Database Server to Use Enterprise Replication

Perform this step only if you plan to use Enterprise Replication with your database server (Version 7.22 or later).

Before you can activate Enterprise Replication, you need to update the **sqlhosts** file or registry key for each database server that participates in Enterprise Replication.

To activate Enterprise Replication for the first time

1. Bring your database server off-line.
2. Define a **group name** for Enterprise Replication.
3. Assign a **group identifier** to the group and include this group ID in the **sqlhosts** option key or column
4. Bring your database server on-line.

5. For upgrades to Versions 7.22 to 7.24, verify that the SNMP service is installed on the host and that the SNMP subagents are registered.

In Dynamic Server 7.30, you need SNMP only if you plan to use the Enterprise Replication Monitoring Program. (Version 7.31 does not include the Enterprise Replication Monitoring Program.)

6. Use the Replication Manager to define each database server for replication.

This step starts Enterprise Replication.

When you upgrade to Dynamic Server 7.3x, the conversion upgrades the **syscdr** database and the **sysmaster** and **sysutils** databases and system tables to Version 7.3x.

When you upgrade to a Version 7.3x database server from a Version 7.2x database server, the Enterprise Replication state is the same in Version 7.3x as it was in Version 7.2x.

For information on how to configure Enterprise Replication, refer to the *Guide to Informix Enterprise Replication*. For information on how to use SNMP, refer to the *Informix SNMP Subagent Guide*.

WIN NT

Install and Configure SNMP

In Dynamic Server 7.30, if you plan to use the Enterprise Replication Monitoring Program, you must install SNMP. Enterprise Replication in Dynamic Server 7.30 does not require SNMP. The SNMP management tools use the **OnSnmpSubagent** to respond to queries.

For Informix database servers Version 7.22 through Version 7.24, if you use Workgroup Replication on Windows NT, the **onsnmp** utility requires Windows NT SNMP. The installation and upgrade program checks the registry for the SNMP master agent. If the master agent was not installed, the program displays a warning message but does not configure the registry for Workgroup Replication or **onsnmp**. If you later choose to install the SNMP master agent, you must run the `%INFORMIXDIR%\bin\inssnmp.exe` command-line utility to install the SNMP subagents. You do not need to reinstall the database server.

UNIX

Bring the Target Database Server On-Line

When you bring your target database server on-line for the first time, bring it first to quiescent mode and then to on-line mode.

Execute the following command to bring your database server from off-line mode to quiescent mode:

```
oninit
```

After the database server is in quiescent mode, check the message log for status messages.

Important: If you note problems in the message file, solve the problems before you continue to the next step.

Execute the following command to change your database server mode from quiescent mode to on-line mode:

```
onmode -m
```

The **sysmaster** and **sysutils** databases are created once your database server is brought on-line.

Warning: The logical logs continue to fill with the transactions that result from the creation of the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, the database server halts with a “Logs Full” error. Thus, you must back up the logical logs.

Use ON-Bar, **ontape**, or ON-Archive to back up logical logs. ♦

The setup program for Windows NT brings your database server on-line automatically. ♦

If you customized the database server environment, you can bring down and restart the database server using IECC. When you restart your database server, the changes to the configuration parameters and environment variables take effect.

You can start the database server with or without the IECC interface. For more information, refer to your *Installation Guide*.



Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement according to the recommended procedure in the *Informix Guide to SQL: Syntax*. UPDATE STATISTICS updates the information that your database server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Verify the Integrity of the Data

Before you allow users to access the databases, use the **oncheck** utility to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. For more information, see [“Verify the Integrity of the Data” on page 11-24](#).

Check Space Requirements for BYTE and TEXT Data

If you are migrating from OnLine Dynamic Server 6.0 or 7.1, the space required for BYTE or TEXT data might have changed. Run **oncheck -cD** on the tables that have BYTE or TEXT data. This **oncheck** command modifies the internal bitmaps to show any changes in space availability. It also shows warning messages when a bitmap has changed. For more information, refer to [“Changes to BlobSpace Requirements in OnLine Dynamic Server 7.10.UD1” on page 11-17](#).

Make an Initial Backup of the Target Database Server

Use your database server backup utility (ON-Bar, ON-Archive, or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of your database server. For more information about making a backup, refer to your *Archive and Backup Guide* or your *Backup and Restore Guide*.



Important: Do not restore the backed-up logical logs from the earlier version of your database server to the newer version of your database server.

WIN NT

Verify the Access Path of the SQL Statements

Use the SET EXPLAIN statement to verify that the access path of your SQL statements did not change when you migrated to your target database server. If you have SET EXPLAIN output from the source database server, run SET EXPLAIN for your target database server. Compare the SET EXPLAIN output from both the source and target database servers. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

For Windows NT, the SET EXPLAIN output filename is %INFORMIXDIR%\sqlexpln\username.out. ♦

If the SET EXPLAIN output file shows that a different access path was used, complete the following steps:

1. Check the **OPTCOMPIND** environment variable or configuration parameter.
2. Check the **DBSPACETEMP** environment variable or configuration parameter to ensure that adequate temporary dbspaces are defined. You might need to define more temporary dbspaces.
3. Analyze the query access paths and modify the schema to improve the performance if necessary.

Complete Migration

The first time your database server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build is complete before you allow users to access the database server. After you complete a level-0 backup and you ensure that client users can access data on your database server, the migration process is complete.

Once you successfully migrate to Dynamic Server 7.3, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between your earlier version and your new version. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on topics related to performance, refer to your *Performance Guide*.

Change Database Server Definitions

Use IECC if you want to change the **informix** user password, specify a different computer as the shared-server computer, or edit database server definitions (the database server name, TCP/IP host names, network types, and the service names). For example, you might specify a different shared-server computer if you have migrated the database server to a new computer or connected a single client to several database servers.

WIN NT

The installation program automatically updates the Windows NT registry information. ♦

For information on how to use IECC to configure database servers, see the *Informix Enterprise Command Center User Guide*. For information on connectivity, see your *Administrator's Guide*.

GLS

Migrating to a GLS Locale

If you want to migrate your database server to a non-English GLS locale, set the DB_LOCALE and CLIENT_LOCALE environment variables before you open the database in Version 7.3. If your previous database server version used Native Language Support (NLS), replace the NLS environment variables, such as **COLLCHAR**, with GLS environment variables. For information on how to work with locales and how to set GLS environment variables, see the *Informix Guide to GLS Functionality* and [Chapter 18, “Converting to GLS.”](#)

Reverting to an Earlier Version of OnLine Dynamic Server

This section covers the changes that occur when you revert to an earlier version of a 7.x through 4.1 database server. You can revert to and from the following versions of database servers.

Revert To	Revert From
Dynamic Server, Version 7.30	Dynamic Server, Version 7.31
OnLine Dynamic Server, Version 7.2x	Dynamic Server, Version 7.3x Later version of OnLine Dynamic Server, Version 7.2x
OnLine Dynamic Server, Version 7.10.UD1 through Version 7.14	Dynamic Server, Version 7.3x Later version of OnLine Dynamic Server
OnLine Dynamic Server, Version 6.0	Dynamic Server, Version 7.3x Later version of OnLine Dynamic Server
OnLine, Version 5.0x	Dynamic Server, Version 7.3x OnLine Dynamic Server, Version 6.0 through Version 7.2x
OnLine, Version 4.1	Dynamic Server, Version 7.3x OnLine Dynamic Server, Version 6.0 through Version 7.2x

This section describes the steps to revert to an earlier version of your database server. When you revert, you must consider changes in the definitions of configuration parameters and environment variables.

UNIX

Follow the preparatory steps, described in [“Preparing for Migration” on page 11-19](#) and then complete the following steps:

1. Remove unsupported SQL features.
1. Remove fragmented tables (for reversion to Version 6.0). ♦
2. Save system catalog information (for reversion to Version 7.x or Version 6.0).
3. Stop and uninstall Enterprise Replication or Workgroup Replication, if it is installed (for reversion from Version 7.22 or later).
4. Uninstall the database server and the administration tools.
5. Modify In-Place ALTER TABLE (for reversion from Version 7.24 and later).
6. Close all transactions and put the source database server in quiescent mode.
7. Verify the integrity of the data.
8. Back up your database server.
9. Run the reversion utility (**onmode -b**).
10. Remove GLS features (for reversion to Version 7.x or Version 6.0).
11. Modify configuration parameters.
12. Reset environment variables.
13. Modify the **sqlhosts** file (for reversion to Version 7.x or Version 6.0). ♦
14. Reinstall the earlier version of the database server.
15. Bring the target database server online.
16. Rename ON-Archive files (for reversion to Version 6.0).
17. Restore system catalog information (for reversion to Version 6.0). ♦
18. Verify the integrity of the data.
19. Back up the target database server files.
20. Return the target database server to on-line mode.
21. Complete reversion.

UNIX

UNIX



Warning: On Windows NT, you need to uninstall Enterprise Replication Manager and your database server and then reinstall the old version of your database server. You cannot have two versions installed concurrently.

UNIX

Remove Unsupported SQL Features

Before you revert, you must remove SQL features that the earlier version of your database server does not support. See the “New Features of This Product” section in the appropriate version of the *Informix Guide to SQL: Syntax*.

Remove Fragmented Tables (Version 6.0 Only)

If you are reverting to Version 6.0, you must change all fragmented tables back into unfragmented tables. For information on how to defragment tables, see your *Administrator's Guide*.

Save System Catalog Information

If your current database server instance uses secure-auditing masks or ON-Archive, and you want to preserve the associated catalog information, you must unload these system catalog tables before you continue. Execute the following command to unload the system catalog tables:

```
$INFORMIXDIR/etc/smi_unld
```

When the **smi_unld** utility finishes unloading the information, the utility displays instructions for reloading the information. *Save these instructions.*

After you complete the reversion and initialize your database server, you can reload the data that you preserved. Follow the instructions given with the **smi_unld** utility for reloading the information. Typically, you should execute the following command:

```
$INFORMIXDIR/etc/smi_load $INFORMIXDIR/etc/
```

Stop Enterprise Replication or Workgroup Replication (Version 7.22 or Later)

Skip this section if Enterprise Replication or Workgroup Replication is not installed on your system.

To revert to an earlier version if Enterprise Replication or Workgroup Replication is active

UNIX

1. Stop Enterprise Replication or Workgroup Replication.
2. For altered tables with CRCOLS, issue the command:

```
alter table drop CRCOLS
```

◆
3. If you are reverting from Dynamic Server 7.3x, modify any SQL SELECT statements that are larger than 255 bytes in the **partdef** table. These SELECT statements are truncated during reversion so that they fit into the table definition of the source version. Fix any SQL SELECT statements that might have been truncated before making Enterprise Replication active again.
4. Execute the **onmode -b** command to revert to the earlier version of your database server.



Warning: If you try to revert to a previous version of the database server while Enterprise Replication is active, the reversion will fail.

5. After you bring up a 7.2x database server, type the following command to change the Enterprise Replication state to ACTIVE:

```
start_cdr
```

To revert to an earlier version if Enterprise Replication or Workgroup Replication is inactive

UNIX

1. In this situation, Enterprise Replication was previously active on this database server. For altered tables with CRCOLS, issue the command:

```
alter table drop CRCOLS
```

◆

2. If you are reverting from Dynamic Server 7.3x, modify any SQL SELECT statements that are larger than 255 bytes in the **partdef** table. These SELECT statements are truncated during reversion so that they fit into the table definition for the older version. Fix any SQL SELECT statements that might have been truncated before making Enterprise Replication active again.
3. Execute the **onmode -b** command to revert to the earlier version of the database server. The **syscdr** database is dropped during reversion.

For more information, see the *Guide to Informix Enterprise Replication*.

Uninstall the Database Server and Informix Enterprise Command Center

Uninstall both the database server and the administration tool, IECC.



Warning: Do not check **Remove all OnLine databases, supporting files and all database information**. If you check this option, your configuration, dbspaces, and database information will be lost, making reversion impossible.

For more information, see the *Informix Enterprise Command Center Installation Guide*.

Modify In-Place ALTER TABLE

Upgrading to OnLine Dynamic Server 7.24 or later versions of the database server occurs automatically when you bring up the target database server. However, reverting to an earlier version of the database server from Version 7.24 or later is not possible if outstanding In-Place ALTER TABLE statements exist. An In-Place ALTER TABLE statement is outstanding when data pages exist with the old definition.

If you attempt to revert to a previous version, the code checks for outstanding alter operations and lists any that it finds. You need to update every row of each table in the outstanding alter list with an ALTER TABLE version and then perform the reversion.

If an In-Place ALTER TABLE statement was performed on a table, you can upgrade the older version to the latest version by running a test UPDATE statement.

For example, run the following test UPDATE statement:

```
update tabl set column1 = column1
```

For more information on In-Place ALTER TABLE, see your *Performance Guide*.

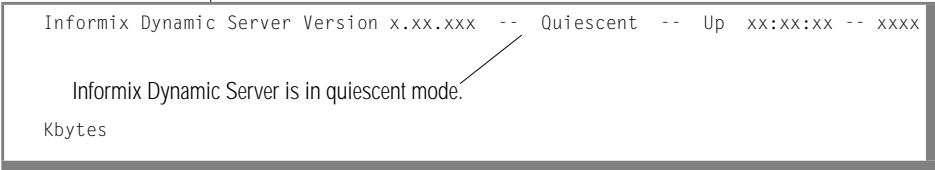
Close All Transactions and Put the Source Database Server in Quiescent Mode

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX platforms. On Windows NT, you must be a member of the **Informix-Admin** group.
3. Execute the **onmode -sy** command.
4. To verify the mode of your database server, execute the **onstat -** command.

The first line of the **onstat** output contains the status of your database server. [Figure 11-6](#) shows that the database server is in quiescent mode.



```
Informix Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx -- xxxx
Informix Dynamic Server is in quiescent mode.
Kbytes
```

Figure 11-6
Example of onstat
Status Line

5. Execute the following command to force a new logical log:

```
onmode -l
```
6. Execute the following command to force a checkpoint:

```
onmode -c
```

Important: Monitor your log to verify that all commands were executed properly and to check for inconsistencies prior to migration.



Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

Figure 11-7 lists the **oncheck** commands that verify data integrity. Informix recommends that at a minimum you run the **-cr** and **-ce** options.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 11-7
*Commands for
Verifying the Data
Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

Back Up the Source Database Server

Informix recommends that you make a level-0 backup of each database server that you plan to migrate. Use ON-Bar, ON-Archive, or the **ontape** utility to perform the backup.

The Windows NT environment does not support ON-Archive. ♦

For information about how to perform a backup, refer to your *Archive and Backup Guide*.

Warning: Backups that you perform under older versions of your database server are not compatible with the newer version. Do not try to restore these backups to the newer version.

WIN NT



Run the Reversion Utility

The database server must be running when you execute the reversion utility. You must use this reversion utility to restore compatibility before users can access the data with the earlier version. The **onmode** utility does not revert changes made to the layout of the data that do not affect compatibility.

To revert to an earlier version of the database server

1. Execute the reversion utility to revert the database, where *version_number* is the earlier database server version (see [Figure 11-8](#)):

```
onmode -b version_number
```
2. Drop the **sysutils** database if the earlier version does not support ON-Bar.

Version 7.22 and later database servers support ON-Bar.

The reversion utility forcibly removes all users and shuts down the database server. After the reversion is complete, the database server ID is off-line. For more information about the **onmode -b** command, refer to [Chapter 24](#), “Using the **onmode** Utility.”

Figure 11-8
Reverting to an Earlier Version of the Database Server

Revert To	Revert From	Command
Version 7.22 to Version 7.24/NT/UNIX	Version 7.3x/NT/UNIX	onmode -b 7.24, onmode -b 7.23, or onmode -b 7.22
Version 7.22/NT/UNIX	Version 7.24/NT/UNIX	Reversion is automatic
Version 7.20.UC1 and prior to 7.30UC1/NT/UNIX	Version 7.22/NT/UNIX	onmode -b 7.2
Version 7.10.UD1 and prior to 7.20.UC1/UNIX	Version 7.2 through 7.22/UNIX	onmode -b 7.1.UD1
Version 7.10.UC1 and prior to 7.10.UD1/UNIX	Version 7.2 through 7.22/UNIX	onmode -b 7.1

(1 of 2)



Revert To	Revert From	Command
Version 7.10.UC1 and prior to 7.10.UD1/UNIX	Version 7.10.UD1 through 7.14/UNIX	onmode -b 7.1
Version 6.0xx except Version 6.0ALS/UNIX	Version 7.20 through 7.22/UNIX	onmode -b 6.0
Version 6.0/UNIX/ALS	Version 7.20 through 7.22/UNIX	onmode -b 6.0A
Version 7.10.UD1/UNIX	Version 7.12 through 7.14/UNIX	Reversion is automatic
The latest maintenance Version 5.0x/UNIX	Version 6.0/UNIX	onmode -b 5.0

(2 of 2)

***Tip:** The **onmode -b** command also rebuilds the user-table indexes automatically.*

Reversion to Version 6.0 from Version 7.3x performs the following actions:

- Verifies that none of the existing tables or indexes are fragmented
- Drops the **sysmaster** database

Remove GLS Features

Skip this step if your database server uses the default English locale (**en_us.8859-1**). To revert the database server to Native Language Support (NLS) or Asian Language Support (ALS) from GLS, set the appropriate NLS or ALS locales and environment variables. For information on working with locales, see Section VI, “[Locale Changes](#),” and the *Informix Guide to GLS Functionality*.

Modify Configuration Parameters

Dynamic Server 7.3x uses configuration parameters that did not appear in earlier versions of the database server. You might want to remove or modify these configuration parameters. For more information, see “[New Configuration Parameters in Dynamic Server 7.3](#)” on page 11-7.

Configuration Changes for OnLine Dynamic Server 7.22

Remove the following configuration parameters from your **ONCONFIG** configuration file:

- CDR_LOGBUFFERS
- CDR_EVALTHREADS
- CDR_DSLOCKWAIT
- CDR_QUEUEMEM
- CDR_NIFUSEHELP
- CDR_NIFMEMS
- SCDR_NIFQUEUES

Version 7.22 changed the default value of the **ALARMPROGRAM** configuration parameter.

You might need to revise the values in your **ONCONFIG** file.

Configuration Changes for OnLine Dynamic Server 7.12 and 7.22

You must add the **ONLANGMAP** configuration parameter to your **ONCONFIG** configuration file before you start OnLine Dynamic Server, Version 7.12. **ONLANGMAP** takes the value **en_US-English**. For information on **ONLANGMAP**, see the documentation notes for **INFORMIX-OnLine Dynamic Server for Windows NT 3.51, Version 7.12.TC2**.

If you are reverting to Version 7.12, remove the **HETERO_COMMIT** configuration parameter from your **ONCONFIG** file.

You might need to revise the values in your **ONCONFIG** file.

Configuration Changes for OnLine Dynamic Server 7.10.UD1 Through 7.14

If you are reverting to Version 7.10.UD1 through Version 7.14, remove the following parameters from your **ONCONFIG** configuration file:

- CHUNKS
- DBSPACES
- HETERO_COMMIT
- TBLSPACES
- TRANSACTIONS
- USERTHREADS

WIN NT

Version 7.2x changed the definition of the following two configuration parameters. You might need to revise the values in your configuration file.

- BUFFERS
- MAX_PDQPRIORITY

Version 7.2x removed the following configuration parameters. If you do not restore the parameters to your **ONCONFIG** file, OnLine Dynamic Server uses the default values. You can add these parameters to your **ONCONFIG** file.

- BUFFSIZE
- PDQPRIORITY

Configuration Changes for OnLine Dynamic Server 7.10.UC1

If you are reverting to Version 7.10.UC1, remove the following parameters from your **ONCONFIG** configuration file:

- LBUPRESERVE
- ONDBSPACEDOWN
- OPCACHEMAX

Version 7.10.UD1 and Version 7.2 changed the definition of the following configuration parameters. You might need to revise the values in your configuration file.

- BUFFERS (changed in 7.10.UD1)
- LOCKS (changed in 7.2)
- MAX_PDQPRIORITY (changed in 7.10.UD1)
- OPTCOMPIND (changed in 7.2)

UNIX

Version 7.10.UD1 removed the following configuration parameters. If you do not restore the parameters to your **ONCONFIG** file, OnLine Dynamic Server uses the default values. You should restore these parameters to your **ONCONFIG** file.

- CHUNKS
- DBSPACES
- PDQPRIORITY (removed in Version 7.20)
- TBLSPACES
- TRANSACTIONS
- USERTHREADS

Version 7.10.UD1 moved the following audit configuration parameters into the audit configuration file (**\$INFORMIXDIR/aaodir/adtcfg.std**). If you use the ON-Audit utility, restore these configuration parameters to your **ONCONFIG** file.

- ADTPATH (UNIX only)
- ADTSIZE (UNIX only) ♦
- ADTERR
- ADTMODE

Configuration Changes for OnLine Dynamic Server 6.0

If you are reverting to Version 6.0, make the changes specified in [“Configuration Changes for OnLine Dynamic Server 7.12 and 7.22”](#) and [“Configuration Changes for OnLine Dynamic Server 7.10.UC1”](#) on [page 11-46](#). In addition, remove the following parameters from your **ONCONFIG** configuration file:

- | | |
|------------------|-------------------|
| ■ ALARMPROGRAM | ■ DS_TOTAL_MEMORY |
| ■ DATASKIP | ■ MAX_PDQPRIORITY |
| ■ DS_MAX_QUERIES | ■ OPTCOMPIND |
| ■ DS_MAX_SCANS | ■ PDQPRIORITY |

Important: Use the same values for your target database server for **ROOTPATH**, **ROOTSIZE**, and **ROOTOFFSET** in your **ONCONFIG** configuration file that you used for your source database server.



Before you revert to OnLine Dynamic Server 6.0, you also need to free any resources that you allocated beyond Version 6.0 limits. Observe the following limits:

- The number of page-cleaner threads is less than 33.
- The number of LRU queues is less than 32.

You can set the maximum number of page-cleaner threads with the CLEANER configuration parameter and the maximum number of LRU queues with the LRUS configuration parameter.

Reset Environment Variables

Reset the environment variables to values that are appropriate for your version of your database server. For information on environment variables, refer to the *Informix Guide to SQL: Reference*.

NLS

Environment Variable Changes for OnLine Dynamic Server 7.10.UD1 Through 7.14

OnLine Dynamic Server, Version 7.10.UD1, through OnLine Dynamic Server, Version 7.14, support NLS but do not support GLS. When you revert, delete the following environment variables:

- | | |
|-----------------|-----------------|
| ■ CC8BITLEVEL | ■ GL_DATE |
| ■ CLIENT_LOCALE | ■ GL_DATETIME |
| ■ DBCENTURY | ■ NODEFDAC |
| ■ DBFLTMASK | ■ ONPLOAD |
| ■ DBONPLOAD | ■ PLCONFIG |
| ■ DB_LOCALE | ■ SERVER_LOCALE |
| ■ ESQLMF | ■ THREADLIB |
| ■ GLS8BITFSYS | |

For reversion from Version 7.12, add the following environment variables:

- **DBNLS 1**
- **COLLCHAR**
- **LANG**

Environment Variable Changes for OnLine Dynamic Server 7.10.UC

When you revert to Version 7.10.UCx, do not use the following environment variables:

- **INFORMIXOPCACHE**
- **INFORMIXSQLHOSTS**
- **NODEFDAC**

The recommended settings of the following environment variables changed between Version 7.1x and Version 7.2x. You might need to reset their values.

- **OPTCOMPIND**
- **PSORT_NPROCS**

Environment Variable Changes for OnLine Dynamic Server 6.0

When you revert to Version 6.0, make the environment variable changes that are specified for Version 7.1x. In addition, do not use the following environment variables:

- **DELIMIDENT**
- **FET_BUF_SIZE**
- **PDQPRIORITY**

The **PDQPRIORITY** environment variable and the SQL **SETPDQPRIORITY** statement were introduced after Version 6.0. The **PDQPRIORITY** environment variable does not cause problems if you leave it set for Version 6.0, but it might cause confusion. You must remove the **SETPDQPRIORITY** statement from your Version 6.0 applications.

UNIX

Modify the sqlhosts File

Version 7.10.UD1 introduced the following enhancements to the **sqlhosts** file. If you use any of these enhancements, you must modify your **sqlhosts** file before you run Version 7.10.UCx or Version 6.0.

- Host-name length of 256 characters
- INFORMIXSQLHOSTS environment variable
- Stream pipes
- Explicit addressing for TCP/IP
- Options field

Reinstall the Earlier Version of the Database Server

Reinstall the earlier version of the database server in the same directory as the files for the previous version.

For more information on installation, refer to your *Installation Guide*.

Warning: Do not select **Copy all files and reconfigure the product**. If you select this option, your configuration and database information will be lost.



Bring the Target Database Server On-Line

Execute the following command to bring your database server to quiescent mode:

```
oninit
```

The database server initializes the shared memory and builds the **sysmaster** database. After the **sysmaster** database is built, the reversion process is complete.

Important: You must add the **ONLANGMAP** parameter to the **ONCONFIG** file before you start OnLine Dynamic Server 7.12 or earlier. **ONLANGMAP** takes the value **en_US-English**. The **ONLANGMAP** configuration parameter ensures backward compatibility with non-NLS locales on Windows NT. If the database server fails on install, just restart it.



WIN NT

To start a Version 7.12 database server on Windows NT

1. From the Windows NT **Main** program group, double-click the **Control Panel** icon.
2. Double-click the **Services** icon.
3. Select **INFORMIX-OnLine Dynamic Server** from the **Services** list box.
4. Click **Start**.

The Services dialog box displays the status of the database server.

If you revert to Version 7.22 on Windows NT, you can start the database server from IECC. ♦



Warning: If you start the database server with the **-iy** parameters after the first time, it will overwrite the existing root dbspace unless you first change the **ROOTPATH** parameter in the **ONCONFIG** file. If the database server overwrites the existing root dbspace, it destroys the information that the root dbspace contains, including information about any databases that you created. Consequently, you must then restore the databases from backup tapes.

UNIX

Rename ON-Archive Files (Version 6.0 Only)

The names of the ON-Archive error message and help files changed between Version 6.0 and Version 7.10. For the correct names for these files, refer to your *Archive and Backup Guide*, Version 6.0.

UNIX

Restore System Catalog Information (Version 6.0 Only)

After you initialize your database server, you might need to restore system catalog information. Follow the instructions in [“Save System Catalog Information” on page 11-38](#).

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

Figure 11-9 lists the **oncheck** commands that verify data integrity. Informix recommends that at a minimum you run the **-cr** and **-ce** options.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 11-9
*Commands for
Verifying the Data
Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

Back Up the Target Database Server Files

After you complete the reversion, Informix recommends that you make a level-0 backup. Use your preferred backup administration utility, ON-Bar, ON-Archive, or **ontape**, to make the backup. For information about how to make a backup, refer to your *Archive and Backup Guide*.



Important: Do not overwrite the tapes that you used to back up your database server.

Return the Target Database Server to On-Line Mode

To bring the database server on-line, execute the following command:

```
onmode -m
```

Complete Reversion

To complete the reversion, ensure that client users can access data on the earlier version of the database server.

Reverting to OnLine Dynamic Server 7.10.UCx from OnLine Dynamic Server 7.10.UDI Through 7.14

OnLine Dynamic Server 7.10.UCx is not compatible with later 7.1x versions of the product. For example, if you want to revert to Version 7.10.UCx from Version 7.12, you must run the **onmode -b** command to restore the data to a form that is compatible with the earlier version.

Versions 7.10.UD1 through 7.14 of OnLine Dynamic Server are compatible with each other, so you do not need to use **onmode -b** to migrate the data.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

[Figure 11-10](#) lists the **oncheck** commands that verify data integrity. Informix recommends that at a minimum you run the **-cr** and **-ce** options.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 11-10
Commands for
Verifying the Data
Integrity

For information about **oncheck**, refer to your *Administrator's Guide*.



Back Up the Target Database Server

After you complete the reversion, Informix recommends that you make a complete backup. Use the **ontape** utility to make the backup. For information about how to use **ontape**, refer to your *Archive and Backup Guide*.

Important: *Do not overwrite the tapes that you used to back up your database server.*

Complete Reversion

To complete the reversion, ensure that client users can access data on the earlier version of your database server.

Migrating Between a 7.x or 6.0 Database Server and a 5.x or 4.1 Database Server

In This Chapter	12-3
Preparing for Migration	12-3
Migration Guidelines.	12-4
Changes in OnLine Dynamic Server 6.0	12-4
Major Changes in OnLine Dynamic Server 6.0.	12-5
Changes to Database Utilities	12-6
Operating-System Configuration Issues	12-6
Estimating the Size and Number of Shared-Memory Segments	12-7
Configuring Semaphore Parameters	12-10
Configuring the Number of Open File Descriptors	12-12
Installing Operating-System Updates.	12-12
Disk-Utilization Issues	12-12
Accommodating System Catalog Indexes	12-13
Accommodating the sysmaster Database	12-14
Locating Temporary Files and Tables	12-15
Accommodating Data Distributions	12-15
Accommodating User-Table Indexes	12-16
Accommodating the Upgrading of User-Table Indexes.	12-18
Managing Secure-Auditing Log Files	12-20
Database Server Configuration Issues	12-20
Changing Environment Variables	12-20
Revising Configuration Parameter Values	12-20
Increasing Logical-Log Space	12-21
Planning for Additional Shared-Memory Usage	12-21
Saving Pre-Existing sysmaster Databases	12-22
Configuring Secondary Database Servers for Data Replication	12-22
Managing Backups	12-22
Making Performance Comparisons	12-23

Storage-Manager Validation and Installation.	12-23
Upgrading to Dynamic Server 7.3x or OnLine Dynamic Server	
from OnLine 5.0 or 4.1	12-24
Install the Latest Maintenance Release for the Current Version	
of OnLine	12-24
Capture Configuration and Chunk-Layout Information	12-25
Close All Transactions and Make a Final Backup of	
the Source Database Server.	12-26
Bring OnLine 5.0 or 4.1 Off-Line	12-28
Install and Configure the Target Database Server	12-29
Reconfigure the UNIX Operating System	12-31
Bring the Target Database Server On-Line.	12-32
Upgrade User-Table Indexes	12-33
Move Tables to Another dbspace	12-34
Use the oncheck Utility to Upgrade Indexes.	12-35
Use UPDATE STATISTICS to Upgrade Indexes	12-35
Use DROP INDEX and CREATE INDEX to	
upgrade Indexes	12-36
Verify the Integrity of the Data	12-37
Make an Initial Backup of the Target Database Server.	12-37
Add rowid Columns to Fragmented Tables	12-37
Complete Migration	12-38
Convert a Database to a Different Locale	12-38
Reverting to OnLine 5.0 or 4.1	12-39
Save Copies of the Current Configuration Files	12-39
Remove Database Server Users	12-40
Verify the Integrity of the Data	12-40
Back Up the Source Database Server	12-40
Remove Features That Later Versions Introduced	12-40
Run the Reversion Utility	12-41
Prepare the TBCONFIG Configuration File	12-42
Reset Environment Variables	12-43
Bring Up OnLine 5.0 or 4.1 in Quiescent Mode	12-43
Verify the Integrity of the Data	12-43
Back Up OnLine 5.0 or 4.1	12-44
Return Online 5.0 or 4.1 to On-Line Mode.	12-44

In This Chapter

This chapter describes how to migrate between Dynamic Server, Version 7.3x, or OnLine Dynamic Server, Version 7.2x, Version 7.1x, or Version 6.0, and OnLine, Version 5.0x or Version 4.1.

For new features in Dynamic Server 7.3x, refer also to [Chapter 11, “Migrating Between Versions of Dynamic Server 7.3 and OnLine Dynamic Server 7.x and 6.0.”](#)

Preparing for Migration

To prepare for migration between a 7.x or 6.0 database server and a 5.x or 4.1 database server, you need to understand the Informix guidelines for migration. You also need to know about any new features that might affect migration.

This section describes the planning and preparations required to migrate to a Version 6.0 or later database server because of the changes that occurred between Version 5.0 and Version 6.0.



Tip: *This chapter discusses migration only on UNIX. For information about migration on Windows NT, see [Chapter 14, “Migrating Between 7.x Database Server Options and Editions.”](#)*

Migration Guidelines

When you migrate from one version of a database server to another, Informix recommends that you consider the following guidelines:

- Review the release notes for the version of your database server for information about new features, installation, and fixes to problems. Modify applications as needed.

The release notes are in `$INFORMIXDIR/release/en_us/0333`.

- Retain both versions of the Informix product software on disk (if you have enough disk resources).
- Check the documentation notes for information about features not covered in the manuals.
- Retain the installation media from both versions of the Informix product software.
- Make a level-0 backup of the database server before and after migration.
- For Dynamic Server 7.3x, verify storage-manager validation.
For details, see [“Storage-Manager Validation and Installation” on page 12-23](#).

In addition to the material in this chapter, read the following information in [“Changes in Dynamic Server 7.3 and OnLine Dynamic Server 6.0 to 7.2” on page 11-6](#). For installation and configuration, refer to your *Installation Guide* and your *Administrator’s Guide*.

Changes in OnLine Dynamic Server 6.0

In Version 6.0, OnLine Dynamic Server introduced an architecture that differs greatly from the architecture used in OnLine 4.1 or 5.0x. This new architecture requires significant changes in allocation and disk usage.

Major Changes in OnLine Dynamic Server 6.0

Figure 12-1 describes the major changes in OnLine Dynamic Server, Version 6.0.

Figure 12-1
Changes Introduced in Version 6.0

Area of Change	Comments
Backup-tape format	Backups and logical-log backups made with pre-6.0 tbtape are not compatible with either of the two Version 7.2x tape utilities, ON-Archive and ontape . Informix recommends that you make a backup before you upgrade to Version 7.2x and then make a second backup after you complete the upgrade.
sqlhosts file	The sqlhosts file is mandatory. You must create an sqlhosts file or modify your current sqlhosts file to the format for OnLine Dynamic Server 7.2x.
Utility names	Version 6.0 introduced a new naming convention for the utilities. The OnLine Dynamic Server utilities use the prefix on (for example, oninit) and the INFORMIX-SE utilities use the prefix se (for example, selog).
sysmaster database	When you initialize Dynamic Server 7.3x, a script automatically creates the sysmaster and sysutils databases. You must ensure that at least 1,100 free pages exist in the root dbspace to build this database.
System resources	The Version 6.0 changes include new requirements for system resources such as shared memory, semaphores, and disk space. When you migrate to a later version from OnLine 4.1 or 5.0, you must reconfigure the operating-system kernel.
Index requirements	To accommodate new features such as key-value locking, the indexing scheme requires an additional 1 byte of disk space per index-key entry. You must rebuild all user indexes after you migrate from OnLine 4.1 or 5.0. An index on a table with one million records requires approximately 1 additional megabyte of disk space.



Changes to Database Utilities

Utility names that began with **tb** in pre-6.0 versions begin with **on** in OnLine Dynamic Server 6.0 and later. For instance, **tbcheck** was replaced by **oncheck**. For the complete list of utilities, refer to your *Administrator's Guide*.

Tip: Plan to update all references to **tb*** utilities.

You cannot use utilities that use binary formats to load and unload data between post-6.0 and pre-6.0 versions. To transfer data between post-6.0 versions and pre-6.0 versions, you can use only those utilities that load and unload data in ASCII format. For instance, the **tbload**, **tbunload**, and **tbtape** utilities in Version 4.1 and 5.0 use binary data and you cannot use them to transfer data to Dynamic Server 7.3. The **onload**, **onunload**, and **ontape** utilities in Version 7.3 use binary data and you cannot use them to transfer data to earlier versions. You can use **dbexport** (pre-6.0 version) and **dbimport** (Version 7.3) to transfer ASCII data to a Version 7.3 database server. You can also use the SQL LOAD and UNLOAD statements to transfer data between versions.

Operating-System Configuration Issues

Database servers later than Version 6.0 require system resources in addition to those required in OnLine 4.1 or 5.0. These resources include additional shared-memory segments, additional semaphores, and additional open-file descriptors per process.

The specific tunable parameters and methods that you use to configure these resources into the operating system vary from platform to platform. For more information, consult the machine notes file that is installed with your distribution of your target database server and the configuration instructions for your operating system.

Estimating the Size and Number of Shared-Memory Segments

When you move from Version 4.1 or 5.0, you must recalculate your memory requirements. Database servers later than Version 6.0 make more extensive use of shared memory than OnLine 4.1 or 5.0. For example, in database servers later than Version 6.0, in addition to housing the buffer cache, virtual processors use shared memory to manage user threads and other activities that individual server processes handled in earlier versions. When you upgrade to Version 7.20 or later from OnLine 4.1 or 5.0, add an additional 8 megabytes of shared memory.

From the standpoint of the operating system, virtual memory that previously was allocated to individual server processes in Version 4.1 and 5.0 is now included in the virtual segment attached by Version 7.20 or later.

This new arrangement requires a new method for calculating shared-memory requirements in the target database server that takes the following items into account:

- The *virtual segment*, which is used to manage multiple user threads, data distributions, and other data
- The familiar RSAM or *resident segment*, which is used to manage the buffer cache
- A new *message segment*, which is used to support the shared-memory communication interface



Tip: *Data distributions in the database server provide the query optimizer with statistical information about the contents of columns and tables. For information about data distributions, refer to UPDATE STATISTICS in the “Informix Guide to SQL: Syntax.”*

The overall system requirement for user virtual memory includes all three shared-memory segments as well as the space that is needed to hold process images. However, the system requirement for actual physical memory includes only the resident segment and the working sets from other segments and processes. Thus, the physical memory that the target database server requires is proportional to the resident segment, while the requirement for swap space is proportional to the total amount of shared memory that the database server uses.

After Version 6.0, OnLine Dynamic Server and Dynamic Server 7.3x on UNIX require at least one virtual shared-memory segment. The default size of a shared-memory segment is 8 megabytes.

You can use the following steps to generate a rough estimate for the size and number of shared-memory segments that are required for your instance of your target database server:

1. Estimate the total amount of shared memory that you need to initialize the target database server. You must make separate estimates for each of the three shared-memory segments, as described in the following paragraphs, and add up the total:
 - For an initial estimate of the resident segment size, use the size of shared memory as displayed in the output of **tbmonitor** under OnLine 4.1 or 5.0. Because the resident segment in your target database server does not include big buffers, you can deduct 4 kilobytes for each 100 buffers in the **BUFFERS** parameter of your OnLine 4.1 or 5.0 **tbconfig** file.
 - Depending on your application, an initial estimate for the virtual segment might be as low as 100 kilobytes per user, or as high as 500 kilobytes per user, plus up to 4 megabytes in addition if you intend to use data distributions. You can obtain an estimate of the number of users under your target database server by adding 12 to the value of the **USERS** parameter in your OnLine 4.1 or 5.0 **tbconfig** file. The initial size of the virtual segment corresponds to the **SHMVIRTSIZE** configuration parameter in your target database server configuration file.
 - Use the following formula to estimate the size of the message segment:

$$\text{msgseg} = (10,531 * \text{connections}) + 50,000$$

connections is the number of user sessions that can connect through the shared-memory interface. You can set the number of sessions with the **NETTYPE** parameter in the OnLine Dynamic Server 7.2 **ONCONFIG** file.

After you start your target database server, you can obtain a more precise value for **SHMVIRTSIZE** with **onstat -g mem**. You can then reconfigure shared memory more precisely with the actual value for **SHMVIRTSIZE** reported by this command.

2. A Version 7.3 database server can attach additional shared-memory resources during operation when it performs a large sort or other operation that might require more memory than it has previously acquired. To allow the database server to expand its use of shared memory while it is operating, reserve a suitable margin of shared memory over that which you estimate is necessary to initialize your target database server. The SHMADD parameter in the **ONCONFIG** file specifies the size of a dynamically added segment. If you do not specify a value for this parameter in your **ONCONFIG** file, the database server attempts to attach additional shared memory in 8-megabyte segments.

The SHMTOTAL parameter in the **ONCONFIG** file places an absolute maximum on the amount of shared memory that an instance of your target database server can request. To avoid the risk of exceeding the shared memory provided for a given instance of your target database server, you can set this parameter to indicate the maximum amount of shared memory for that instance. If you set SHMTOTAL to 0 or leave it unassigned, the database server continues to attach additional shared memory as needed until no more virtual memory is available on the system.

3. Estimate the size and number of shared-memory segments that the operating system needs to provide and then modify your kernel.

If your operating system *does not* have a segment-size limit, take the following actions:

- a. Set the maximum-segment-size parameter, typically SHMMAX or SHMSIZE, to the total size that is required for your target database server. Include both the amount of memory that is required to initialize the database server as calculated in step 1 on [page 12-8](#), and the amount of memory that you want to allow for dynamic growth as described in step 2 on [page 12-9](#).
- b. Set the operating-system configuration parameter for the maximum number of segments, typically SHMMNI, to at least 1 per instance of the database server.

If your system *does* have a segment-size limit, take these actions:

- a. Set the maximum segment-size parameter for the operating system, typically SHMMAX or SHMSIZE, to the largest value that your system allows.
- b. Use the following formula to calculate the number of segments for your instance of your target database server:

$$\text{SHMMNI} = ((\text{initial_segment}) / \text{SHMMAX}) + \text{dynamic_segments}$$

initial_segment is the segment size that is required to initialize your target database server.

dynamic_segments is the number of segments that you allow to be added during operation of your target database server.

If a remainder exists, round up to the nearest integer value.

4. If your operating system uses the SHMSEG configuration parameter to indicate the maximum number of shared-memory segments to which a process can attach, set this system-configuration parameter to a value that is equal to or greater than the largest number of segments that you allocated for any one instance of your target database server.

Configuring Semaphore Parameters

The operating-system configuration parameters for semaphores are calculated differently for OnLine Dynamic Server 7.10 or later than for pre-6.0 versions of the database server. On UNIX systems, the SEMMNI parameter gives the number of semaphore sets. Each instance of your target database server requires one set of semaphores for each group of (up to) 100 virtual processors (VPs) that are initialized with the database server, one set for each additional VP that you might add dynamically (while the database server is running), and one set for each group of 100 (or fewer) user sessions that are connected through the shared-memory communication interface. Because the target database server utilities such as **onmode** use shared-memory connections, you must configure a minimum of two semaphore sets for each instance of your target database server: one for the initial set of VPs and one for the shared-memory connections that the database server utilities use.

The SEMMSL operating-system configuration parameter typically gives the maximum number of semaphores per set; set this parameter to (no less than) 100.

On systems that require you to configure a maximum for the total number of semaphores across all sets, typically given by the SEMMNS operating-system configuration parameter, use the following formula to calculate the total required for each instance of your target database server:

$$\text{SEMMNS} = \text{init_vps} + \text{added_vps} + \text{shmem_users} + \text{concurrent_utils}$$

<i>init_vps</i>	is the number of VPs that are initialized with your target database server. This number includes CPU, PIO, LIO, AIO, SHM, TLI, SOC, and ADM VPs. (For a description of these virtual processors, see your <i>Administrator's Guide</i> .) The minimum value for this term is 15.
<i>added_vps</i>	is the number of VPs that you can add dynamically.
<i>shmem_users</i>	is the number of shared-memory connections that are allowed for this instance of your target database server.
<i>concurrent_utils</i>	is the number of concurrent database server utilities that can connect to this instance. Informix suggests that you allow for a minimum of six utility connections: two for onarchive and four for other utilities such as onmonitor , oncheck , and onstat .

For example, if you start a single instance of your target database server with two CPU VPs and 110 shared-memory users, and you intend to add two CPU VPs dynamically as needed, you must include at least five semaphore sets in the SEMMNI parameter: one set for the initial VPs, two sets for the dynamically added CPU VPs, and two sets for the shared-memory connections. You must set the SEMMSL parameter to at least 100. If your system requires a value for the SEMMNS parameter, you must indicate a total of no less than 133 (15 + 2 + 110 + 6).

If your system uses software packages that require semaphores in addition to the ones that your target database server needs, you must include the total number of semaphore sets that are required by both the database server and your other software packages in the SEMMNI parameter. You must set the SEMMSL parameter to the largest number of semaphores per set that any package requires. For systems that require the SEMMNS parameter, you can multiply SEMMNI by the value of SEMMSL to calculate an acceptable value. Or, to arrive at a more precise value for SEMMNS, you can calculate the number of semaphores that are needed for each software package and add those numbers to obtain the total.

Configuring the Number of Open File Descriptors

Some operating systems require you to specify a limit on the number of file descriptors that a process can have open at any one time. You specify this limit with an operating-system configuration parameter, typically NOFILE, NOFILES, NFILE, or NFILES. The number of open file descriptors that each instance of your target database server needs is the number of chunks in your database plus the number of network connections that your database server instance must support.

Installing Operating-System Updates

The target database server might require you to install operating-system updates or *patch releases*. For information about operating-system patches that your target database server installation might require, refer to the machine notes file.

Disk-Utilization Issues

When you upgrade from a pre-6.0 version of the database server, you must allow for increased disk use both during and after the upgrade process. Some disk resources that must be allocated to the root dbspace during the upgrade process can be freed for other uses after the process completes. The additional disk space that is required falls into the following categories:

- Space for the upgrade processes
- Space in each index entry
- Space for data distributions

The initial requirements during the upgrade include 1,100 additional pages in the root dbspace over its size under pre-6.0 versions of the database server. The database server uses these added pages to build the **sysmaster** database. You must also provide additional space in the root dbspace for automatic upgrading of system catalog indexes.

An additional byte per index entry for each user-table index must be allocated to the dbspace in which each corresponding table resides.

If you intend to use data distributions, you must provide enough temporary space to hold the largest table for which you intend to establish a distribution. For information about data distributions, refer to UPDATE STATISTICS in the *Informix Guide to SQL: Syntax*.

The remainder of this section outlines the additional hard-disk requirements for migration to your target database server. You need to examine carefully the amount of disk space left in each dbspace.

The first time that you start up your target database server, the database server performs the following tasks automatically:

- Upgrading of system catalog indexes to your target database server, (occurs at the start of quiescent mode)
- Creation of **sysmaster** and **sysutils** databases (occurs at the start of on-line mode)

You must provide enough space for these actions to take effect. After the database server begins normal operation, you must rebuild the indexes for user tables. You must allocate adequate disk resources for those indexes as well.

Accommodating System Catalog Indexes

The following formula indicates how many additional pages you must provide to accommodate the growth in system catalog indexes for a given database:

$$\text{growth_in_pages} = (\text{Total} * .10) + (\text{Largest} * 1.10)$$

Total is the total number of leaf pages for all system catalog indexes.

Largest is the number of leaf pages in the largest index.

The additional space that is available during the upgrade process must include room for the following items:

- A copy of the largest index plus 10 percent
- 10 percent of the current total of system catalog index pages

The database server requires this amount of additional space to accommodate those brief periods in which the old and new versions of an index both reside on disk. The additional 10 percent allows for the case in which the largest index also is the last one to be upgraded.

You can use the following SQL query within DB-Access or INFORMIX-SQL to determine how many added pages the new system catalog indexes require:

```
UPDATE STATISTICS;  
SELECT ((SUM(leaves) * 0.10) + (MAX(leaves) * 1.10)) sci_added  
FROM sysindexes  
WHERE tabid < 100;
```



Important: You must perform this query from OnLine 4.1 or 5.0.

If sufficient space is not already available in the root dbspace, you must allocate additional chunks or move tables to other dbspaces to make room. You can use the **tbstat -d** command to find the number of free pages in the root dbspace. For details, refer to your *Administrator's Guide*, Version 4.1 or 5.0.

Accommodating the sysmaster Database

The **sysmaster** database is created in the root dbspace and cannot be moved or redirected. The **sysmaster** database contains *pseudotables* for monitoring and real tables to store backup information for ON-Archive. To create the **sysmaster** database, you need up to 1,100 free pages in the root dbspace.

Locating Temporary Files and Tables

Version 6.0 and later versions of the database server let you use the **DBSPACETEMP** configuration parameter or the **DBSPACETEMP** environment variable to specify the location of temporary files and tables in either raw or cooked space in UNIX. When you use Windows NT, you can specify the locations of temporary files and tables in buffered and unbuffered space. (Previous versions of the database server create temporary tables in the root dbspace by default. Version 6.0 and later versions of the database server do not use the **DBPATH** environment variable to locate sort files.) If your computer has at least two hard disks, you might consider mirroring your root dbspace and redirecting the temporary table creation elsewhere. This setup prevents hard-disk failures on the root dbspace from affecting your day-to-day business activities.

Accommodating Data Distributions

If you intend to use data distributions, you must provide adequate space for them in the dbspace that contains the system catalog tables for each database in which they are used. (Use the **UPDATE STATISTICS** statement, described in the *Informix Guide to SQL: Syntax*, to create data distributions.) The following formula gives the maximum amount of space in bytes that might be required for a data distribution on an individual column. Add the results for each column to obtain the total amount of additional space that is needed for the distributions themselves:

$$\text{dist_space} = (\text{ceil}((4 * ((1/d_res) + 1) * (4 + c_len)) / 765) * 1,116) + 2$$

- ceil*** represents a mathematical function that rounds its argument to the next larger integer. On many systems, this function is called *ceil* or *ceiling*.
- d_res*** is the decimal representation of the resolution that is specified in the **UPDATE STATISTICS** statement. The default resolution for a HIGH-mode data distribution is 0.5 percent, or 0.005 in this formula.
- c_len*** is the length in bytes for the column. A **FLOAT** column typically contains 8 bytes; a **CHAR 20** column contains 20 bytes.



Tip: This formula yields the maximum possible size for a distribution that contains a number of overflow entries. A typical distribution with few or no overflow entries uses only 25 percent of the maximum space that this formula projects.

For MEDIUM-mode distributions, you must also provide sort space in the dbspace equivalent to 3,000 rows of the widest table. For HIGH-mode distributions, you must provide space for a complete copy of the largest table for which you want a HIGH-mode data distribution.

For example, if you intend to use a HIGH-mode data distribution with the default resolution of 0.5 percent on a CHAR 20 column, the following calculation shows the maximum space that is needed for that distribution in bytes:

$$(\text{ceil}((4 * ((1/0.005) + 1) * (4 + 20))/765) * 1,116) + 26$$

This formula works out to 29,042 bytes, as the following calculations show:

$$\begin{aligned} &(\text{ceil}((4 * (200 + 1) * 24)/765) * 1,116) + 26 \\ &(\text{ceil}((4 * 201 * 24)/765) * 1,116) + 26 \\ &(\text{ceil}(19,296/765) * 1,116) + 26 \\ &(\text{ceil}(25.223) * 1,116) + 26 \\ &(26 * 1,116) + 26 \\ &29,016 + 26 \end{aligned}$$

If this column appears in a table that contains 100,000 rows of 28 bytes each, you must provide a minimum of approximately 2.8 megabytes of sort space in the dbspace to build this distribution.

For a MEDIUM-mode distribution, calculate the required amount of sort space as follows:

$$\text{sort_space} = 28\text{B} * 3,000 = 84,000\text{B}$$

Accommodating User-Table Indexes

The following formula indicates the number of additional pages necessary to accommodate the growth in user-table indexes for a given database:

$$\text{index_growth_pages} = (\text{total} * 0.10)$$

total is the total number of leaf pages for all user-table indexes.

You might need to add chunks to your existing dbspaces or perhaps add new dbspaces and move tables to that dbspace to provide additional room. You can use the **tbstat -d** command to find out the number of free pages in the current dbspace. For details, refer to your *Administrator's Guide*, Version 4.1 or 5.0.

You can use the following SQL query in DB-Access or INFORMIX-SQL to determine the number of additional pages that user-table indexes need for an entire database:

```
SELECT (SUM(leaves) * 0.1) uti_added
FROM sysindexes
WHERE tabid >= 100;
```

Indexes reside in the same dbspace as the tables to which they refer. Tables can be located in different dbspaces than the databases in which they are managed. If all your tables reside in the current dbspace, the result of this query indicates the number of pages to add to the database. However, if one or more tables reside in separate dbspaces, you must make sure that those dbspaces include enough room for the new indexes.

To find out the dbspace in which each external table resides, use the following SQL query:

```
SELECT tabname, tabid, (partnum / "0x100000") dbspace_num
FROM systables
WHERE tabid >= 100
AND partnum > 0;
```

For each table that resides in a separate dbspace, use the following SQL query to find out how many additional index pages must be added to the estimate for its dbspace. Add the resulting number of pages to the estimate for that dbspace and deduct it from your estimate for the current dbspace, as follows:

```
SELECT (leaves * 0.1) tbl_added
FROM sysindexes
WHERE tabid = alt_tabid;
```

alt_tabid is the tabid (table ID number) of the table in a separate dbspace as returned by the previous query.

Accommodating the Upgrading of User-Table Indexes

The process of upgrading user-table indexes for your target database server to use is not automatic and requires preliminary planning. You upgrade user indexes in the last steps of the migration procedure, after you install the target database server. For more information, refer to [“Use the oncheck Utility to Upgrade Indexes” on page 12-35](#).

You can choose from three different methods to upgrade indexes for a user table. The method that you choose for a given table depends on the size of the table, the degree to which availability of the table is seen as critical, the logging mode of the database, and the time that you can allow for your target database server to stay closed to other users.

You can use the following upgrade methods:

- The **oncheck -cI -y** command
- The UPDATE STATISTICS statement
- The DROP INDEX and CREATE INDEX statements



***Tip:** The time you need to upgrade user-table indexes is proportional to the time you need to rebuild indexes in pre-6.0 versions of the database server. Factors that affect the upgrade time include the number of indexes per table, the size of each index, the number of available CPUs, and the absence or presence of other user activity on the system.*

The **oncheck -cI -y** command upgrades indexes while the database is in quiescent mode. This method is preferred for upgrading large or critical tables. You can also use this method to upgrade an entire ANSI-compliant database in a single operation. Usually, this method is also the easiest method to use for any table or database. However, the database server remains unavailable to other users as long as the **oncheck** utility is running. For more information about the **oncheck** utility, refer to your *Administrator's Guide*.

The UPDATE STATISTICS statement provides added flexibility. You can use it to upgrade indexes while other users are on-line. However, if you execute this statement within a transaction while other users are working, you risk bringing the database server off-line with a long-transaction error. Handle small tables in ANSI-compliant databases one at a time if other users are working on the database server. Do not use this method to upgrade large or multiple tables in ANSI-compliant databases.

Dropping and rebuilding indexes is another upgrade option that you can use in place of UPDATE STATISTICS. Because the UPDATE STATISTICS statement also allows you to generate data distributions, it is generally preferred over dropping and rebuilding indexes with the DROP INDEX and CREATE INDEX statements. For more information about these statements, refer to the *Informix Guide to SQL: Syntax*.

To prepare for upgrading a user index

1. Execute the following query in each database:

```
SELECT tabname
FROM systables
WHERE tabid >= 100
```

2. Determine which method to use for each table in your list.

The following chart gives recommendations for tables depending on their size, importance, the type of database in which they reside, and the urgency with which the database server must be brought back on-line. Use **oncheck** to upgrade large, critical tables. The choice for small, but critical, or large, but noncritical tables depends on which scenario produces the smaller effect: having the database server unavailable or rebuilding indexes while your system is active.

Type of Table	Needed Quickly	Less Urgent
Large, critical	oncheck -cI -y	oncheck -cI -y
Small, critical	oncheck -cI -y	UPDATE STATISTICS
Large, noncritical	oncheck -cI -y	UPDATE STATISTICS
Small, noncritical	UPDATE STATISTICS	UPDATE STATISTICS



Tip: You might be able to use a single method to upgrade the indexes on all or most tables within a database. Both the **oncheck** command and UPDATE STATISTICS statement allow you to use a single command to upgrade the indexes for an entire database. For details, see the “*Informix Guide to SQL: Syntax*.”

Managing Secure-Auditing Log Files

If you intend to use the secure-auditing features of the target database server, be advised that the database server audit-record log files can grow rapidly to take up a significant amount of space in the file system. Be sure to allow adequate space for the file system in which these files reside. You can configure audit records to minimize the effect of secure auditing on disk use. For details, refer to your *Trusted Facility Manual*.

Database Server Configuration Issues

This section provides an overview of database configuration issues involved in the migration process. The following discussions describe only those configuration issues that affect the migration process. For detailed information about the database server configuration parameters, refer to your *Administrator's Guide*.

Changing Environment Variables

Version 6.0 and later versions of the database server include new environment variables that replace those in OnLine 4.1 and 5.0. Environment variable names that began with **TB** in earlier versions begin with **ON** in the database server versions 6.0 or later. For instance, the **TBCONFIG** environment variable is replaced by the **ONCONFIG** variable. The **SQLEXEC** environment variable is not used in Versions 6.0 and later.

When you initialize an instance of your target database server, you must set the **INFORMIXSERVER** environment variable to the dbservername of that instance. Applications must also set the **INFORMIXSERVER** environment variable to gain access to databases that your target database server manages.

Revising Configuration Parameter Values

The database server, Version 6.0 or later, recognizes new minimum values for certain configuration parameters that existed prior to Version 6.0.

As of Version 7.10, the **BUFFERS** parameter indicates the maximum number of buffers for disk I/O. To improve performance, increase the **BUFFERS** and **DBSPACETEMP** values, if necessary.

You must increase the value of the LOCKS parameter to at least 2,000. The database server puts a message in the message log if more locks are needed.

Increasing Logical-Log Space

Make sure that at least 2,000 total log pages are allocated and free for logical logs because the building of the **sysmaster** database requires 1,000 log pages. Informix recommends a 1,000 log-page safety factor. Run **tbstat -l** for your current log-usage status.

Planning for Additional Shared-Memory Usage

The database server architecture, introduced with Version 6.0, combines all the memory that **sqlturbo** processes use in previous versions of the database server into a shared-memory section called the *virtual segment* of shared memory. When you migrate from OnLine 4.1 or 5.0, you must allocate enough shared memory with the SHMVIRTSIZE configuration parameter, in the ONCONFIG file, to accommodate the user threads that separate database server processes serviced in pre-6.0 versions. A reasonable initial estimate for SHMVIRTSIZE is 500 kilobytes for each user thread. Additional space might be required for use with data distributions. Use the SHMVIRTSIZE configuration parameter to specify the amount of shared memory for the database server.

The database server requires three segments of shared memory as opposed to the one segment that was required in pre-6.0 versions of the database server. For information about how to configure shared-memory segments in the operating system, refer to [“Estimating the Size and Number of Shared-Memory Segments” on page 12-7](#).

You use the SHMADD parameter in the ONCONFIG file to set the size of a dynamically added segment. If you do not specify a value for this parameter, the database server attempts to attach additional shared memory in 8-megabyte segments.

The SHMTOTAL parameter in the ONCONFIG file places an absolute maximum on the amount of shared memory that an instance of Dynamic Server 7.3 can request. To avoid the risk of exceeding the shared memory provided for the database server, set this parameter to the maximum amount of shared memory for that instance requires. If you set SHMTOTAL to 0 or leave it unassigned, the database server continues to attach additional shared memory as needed until no more virtual memory is available on the system.

Saving Pre-Existing sysmaster Databases

Pre-6.0 versions of the database server include the **makeps.sql** script, which creates a **sysmaster** database. If this database exists on your system, you can run **dropps.sql** to drop it. If you do not drop the old **sysmaster** database, the upgrade process renames it **sysmaster_pre60**.

Configuring Secondary Database Servers for Data Replication

When you configure a secondary database server for use in data replication, the version of your target database server on the secondary host computer must match that on the primary host computer. Chunk names and offsets must also match between instances of your target database server on the primary and secondary hosts. For more information about data replication, refer to your *Administrator's Guide*.

Managing Backups

The ON-Archive backup-management system, introduced in OnLine Dynamic Server 6.0, and table fragmentation, introduced in OnLine Dynamic Server 7.10, might change the placement of databases and tables on disk.

The ON-Archive menu uses a special termcap file that is located in the **tctermcap** file in **\$INFORMIXDIR/etc**. If your terminal is not listed in this file, you might need to add a new entry to use the menu interface. For more information about backup strategies, table organization, and the **tctermcap** file, refer to your *Administrator's Guide*.

Making Performance Comparisons

Informix recommends that you run and record time- and resource-use statistics for sample queries and other operations to help you compare performance before and after migration to your target database server. You can compare these statistics with equivalent operations you perform after the upgrade to characterize performance enhancement or degradation. The comparison might help you identify database configuration parameters that you can adjust to obtain better performance.

Storage-Manager Validation and Installation

When you upgrade or revert an Informix database server, the storage manager that you used on the source database server might not be validated for the version of the database server to which you are migrating. Verify that Informix has validated the storage manager for the target database server version and platform. If not, you need to install a validated storage manager before you perform backups with the ON-Bar backup and restore system.

When you migrate to a new database server version, install the storage manager before you bring up the database server. That way, if you have automatic log backup set up on the database server, ON-Bar can start backing up the logs when the database server comes on-line.



Warning: *If you migrate Informix Storage Manager (ISM) 1.0 catalogs to ISM 2.0 using the `ism_catalog` utility, the catalogs become corrupted. Once the ISM 2.0 server is restarted after catalog migration, error messages occur in various logs.*

For information on how to install and upgrade the storage manager, see the *Informix Storage Manager Administrator's Guide*.

Upgrading to Dynamic Server 7.3x or OnLine Dynamic Server from OnLine 5.0 or 4.1

The procedure for upgrading to Dynamic Server 7.3 or OnLine Dynamic Server from a pre-6.0 database server on UNIX includes the following steps, which the rest of this section describes:

1. Install the latest maintenance release for the current version.
2. Capture configuration and chunk-layout information in OnLine, Version 4.1 or 5.0.
3. Close all transactions and make a final (level-0) backup under OnLine, Version 4.1 or 5.0.
4. Bring OnLine, Version 4.1 or 5.0, off-line.
5. Install and configure the target database server.
6. Reconfigure the operating system.
7. Bring your target database server on-line.
8. Upgrade user-table indexes.
9. Verify the integrity of the database.
10. Make an initial (level-0) backup under your target database server.
11. Add rowid columns to fragmented tables.
12. Complete migration.

Install the Latest Maintenance Release for the Current Version of OnLine

Informix recommends that you install the latest maintenance release for your current database server version before you migrate to a new version, especially if you are using Version 5.0.

For example, to migrate to Dynamic Server 7.3 or to OnLine Dynamic Server from OnLine 5.03 on UNIX, first install the latest maintenance release for OnLine and then migrate to Dynamic Server 7.3 or OnLine Dynamic Server. Many minor changes to the 5.x versions are also in the 7.x versions.

Capture Configuration and Chunk-Layout Information

Before you can safely configure your target database server, you must capture the configuration and chunk-layout information under OnLine 4.1 or 5.0. You use this information when you configure your target database server.

To copy database server configuration files

1. Log in as user **informix**.
2. Ask all OnLine 4.1 or 5.0 users to exit their applications.
3. Rename or make a copy of the configuration file(s) if the current name appears in the following list:
 - **\$INFORMIXDIR/etc/onconfig**
 - **\$INFORMIXDIR/etc/onconfig.std**
 - **\$INFORMIXDIR/etc/sqlhosts**
 - **\$INFORMIXDIR/etc/tbconfig**
 - **\$INFORMIXDIR/etc/tbconfig.std**
 - **\$INFORMIXDIR/etc/tctermcap**
 - **\$INFORMIXDIR/etc/termcap**

You must save these files because the installation procedure for your target database server overwrites them during the installation. Keep the copies available to use later.

To copy chunk-layout and space-use information

Execute the following command to save a listing of your chunk layout and space use in case you need to refer to it:

```
tbstat -d > $INFORMIXDIR/chunk.layout
```

To add chunks or to move tables to accommodate your target database server disk-use requirements

If you need to add chunks or move tables out of the root dbspace to make room for the **sysmaster** database and system catalog indexes, you can do so at this time. You can also add chunks or move tables in this and other dbspaces to accommodate user-table indexes.

For more information about disk-space requirements in your target database server, refer to [“Disk-Utilization Issues” on page 12-12](#). For information about adding chunks or moving tables, refer to your *Administrator’s Guide*, Version 5.0 or 4.1.

Close All Transactions and Make a Final Backup of the Source Database Server

Communicate to client users how long you expect the database server to be off-line for migration. This procedure prepares the transaction log for migration to your target database server and creates a final backup of the database under OnLine 4.1 or 5.0.

To perform an immediate shutdown

Execute the following command to perform an immediate shutdown:

```
tbmode -k
```

Answer *yes* to all the prompts. This step terminates all database server processes that might still be running.

To shut down the system gracefully

1. Execute the **tbmode -sy** command.
2. Wait for all users to exit.
3. Execute the **tbmode -k** command and then answer *yes* to all the prompts.

Make sure the system has shut down completely before you proceed to the next step.

To initiate a fast recovery

Execute the following command to enter quiescent mode and initiate a fast recovery:

```
tbinit -s
```

The **tbinit -s** option rolls forward all committed transactions, rolls back all incomplete transactions since the last checkpoint, and leaves a new checkpoint record in the log with no open transactions pending. (Refer to your *Administrator’s Guide*, Version 5.0 or 4.1.)

To verify the operating mode

Execute the following command to verify that OnLine 4.1 or 5.0 is in quiescent mode:

```
tbstat -
```

The first line of the **tbstat** output contains the status of OnLine 4.1 or 5.0. [Figure 12-2](#) shows that OnLine 4.1 or 5.0 is in quiescent mode.

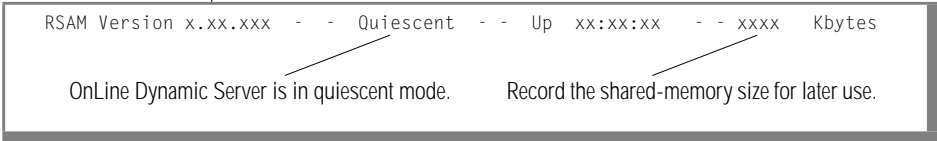


Figure 12-2
Example of **tbstat**
Status Line

To verify the integrity of the data

Before you make a level-0 backup, you might want to verify the integrity of the data.

Use the **tbcheck** utility to verify the integrity of data before you start the backup. You can verify the integrity of the reserve pages, system catalog tables, data, and indexes. (Refer to your *Administrator's Guide*, Version 4.1 or 5.0.) Execute the following commands.

To check	Use the command
Reserve pages	tbcheck -cr
System catalog tables	tbcheck -cc <i>database_name</i>
Data	tbcheck -cD <i>database_name</i>
Indexes	tbcheck -cI <i>database_name</i>

To make a final backup of OnLine 4.1 or 5.0

Use **tbtape** or **tbmonitor** to make a level-0 backup of OnLine 4.1 or 5.0. The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. (For more information on making a level-0 tape backup, refer to your *Administrator's Guide*, Version 4.1 or 5.0.)



Important: Backups that you made under OnLine 4.1 and 5.0 are not compatible with your target database server tape format. Use only the older version of the source database server to restore the backup tape that you made in this step. You cannot use this backup tape with your target database server.

Bring OnLine 5.0 or 4.1 Off-Line

This procedure shuts down the instance of OnLine 4.1 or 5.0.

To shut down the instance of OnLine

Execute the following command:

```
tbmode -k
```

Answer **yes** to all prompts.

To verify that OnLine 4.1 or 5.0 is off-line

Execute the following command to verify that OnLine 4.1 or 5.0 is in off-line mode:

```
tbmonitor
```



Tip: Use **tbmonitor** instead of **tbstat** to verify the operating mode. The **tbstat** utility is not designed to return the operating-mode status when the database server is off-line.

The third line of the DB-Monitor main menu contains the status of OnLine 4.1 or 5.0. The screen in [Figure 12-3](#) indicates that the database server is off-line. (Refer to your *Administrator's Guide*, Version 4.1 or 5.0.)

```
INFORMIX-OnLine:  Status   Parameters  Dbspaces   Mode    Force-Ckpt
Status menu to view INFORMIX-OnLine
-----Off-Line-----Press CTRL-W for Help -----
```

Figure 12-3
DB-Monitor Main
Menu

The database server must be off-line because the older and the newer versions share common files. You cannot install the database server if any of the common files are active. Bring the database server off-line to ensure that all common files are inactive.



Important: Repeat the previous three sections ([pages 12-25 through 12-28](#)) for each database server instance that you want to upgrade.

Install and Configure the Target Database Server

You must be user **root** to install the target database server. Set the **\$INFORMIXDIR** environment variable to the directory where you plan to install the database server.

Follow the directions in your *Installation Guide* and your *Administrator's Guide* to install the target database server. The installation script installs the database server into the **\$INFORMIXDIR** directory specified for user **root**. The installation script does not bring the database server on-line.



Warning: *If you install the target database server in the same directory where the older version of the database server currently resides, the newer version overwrites the older files. If you want to preserve your product files of earlier versions, you must install your target database server in a different directory.*

Before you overwrite the older version, you must take the following precautions:

- If you do not have the original media for the older version, back up the **\$INFORMIXDIR** directory before you install your target database server.
- Copy the configuration file(s) in **\$INFORMIXDIR/etc** to another location on the file system.

When the installation is complete, exit as user **root** and log in as user **informix**.



Important: *The target database server includes networking capabilities that are not present in pre-6.0 versions. These capabilities use networking information from configuration files such as the **ONCONFIG** and **sqlhosts** files to establish communications with application processes. You must ensure that these files are present and contain the needed information. To verify these files, establish a connection from DB-Access to test the database server instance before you proceed with your installation. If your configuration calls for it, you can also test a remote client-server system.*

To set up the configuration file

Copy the standard Dynamic Server 7.3 or OnLine Dynamic Server configuration file to the name of your working configuration file and then edit it. For example, if your configuration filename is **onconfig.1**, you might use the following commands:

```
cd $INFORMIXDIR/etc
cp onconfig.std onconfig.1
vi onconfig.1
```

Refer to the backed-up copy of the OnLine 4.1 or 5.0 configuration file that you saved in [“Capture Configuration and Chunk-Layout Information” on page 12-25](#). Copy the values from matching parameters in the backed-up file to your working configuration file (for example, **onconfig.1**). Modify those values to conform with revised minimums for Dynamic Server 7.3 or OnLine Dynamic Server as described in [“Revising Configuration Parameter Values” on page 12-20](#).

To set environment variables for the target database server

At this point, you must set up the environment variables that both the client applications and your target database server need to communicate. Make sure that the **INFORMIXDIR** and **PATH** environment variables were set during the installation. (For more information, refer to your *Administrator's Guide*.)

Set the **ONCONFIG** environment variable to the name of your target database server configuration file, as the following examples show:

C shell: `setenv ONCONFIG onconfig.1`

Bourne shell: `NCONFIG=onconfig.1;
export ONCONFIG`

Set the **INFORMIXSERVER** environment variable for all users that need to access the target database server. Set this variable to the name that is listed in the **sqlhosts** file and in the **DBSERVERNAME** or **DBSERVERALIASES** parameter of your target database server configuration file, as follows:

C shell: `setenv INFORMIXSERVER dbserver1`

Bourne shell: `INFORMIXSERVER=dbserver1;
export INFORMIXSERVER`

If you use a dbspace or file to store temporary tables, set the DBSPACETEMP configuration parameter or **DBSPACETEMP** environment variable to the name of the dbspace or full pathname of that file.

To modify the sqlhosts file

The target database server requires an **sqlhosts** file. An **sqlhosts.demo** file is included in **\$INFORMIXDIR/etc** as an example of the setup that your target database server requires. If you do not have an **sqlhosts** file already, the installation program renames the **sqlhosts.demo** file to **sqlhosts**. The **sqlhosts** file should include an entry with the following information for each instance of your target database server:

```
dbservername  nettype  hostname  service_name  options
```

You must modify the entries in this file to support your configuration. For more information on how to modify the **sqlhosts** file, refer to your *Administrator's Guide*.



Important: The client application looks for the **sqlhosts** file in the **\$INFORMIXDIR/etc** directory. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the **sqlhosts** file.

Reconfigure the UNIX Operating System

For this step, you need to reconfigure your operating system based on the estimates that you determined in [“Preparing for Migration” on page 12-3](#). You need to provide additional shared memory, additional semaphores, and possibly additional hardware resources such as disk drives. To reconfigure the operating system, follow the directions in the machine notes file included in your target database server distribution and the kernel-configuration instructions for your operating system.

Bring the Target Database Server On-Line

This step brings the database server to quiescent mode. The success of this step depends on adequate operating-system and disk resources, as discussed in [“Preparing for Migration” on page 12-3](#). If the system is not brought up in quiescent mode, you get the following error when you attempt to initialize your target database server and the database server goes off-line:

```
Open transaction detected when changing log versions.
```

To bring the target database server from off-line to quiescent mode

To bring your target database server from off-line to quiescent mode, execute the following command:

```
oninit -s
```

Execute the **onstat -m** command to check the message log for the status of the mode change and to create the **sysmaster** database.

The system automatically creates the **sysmaster** database when the target database server is brought on-line. If the system fails to create this database, the root dbspace might not include the additional 550 pages that are needed for the upgrade. Return to [“Install and Configure the Target Database Server” on page 12-29](#), provide enough space in the root dbspace, and repeat the steps in this section. If the **sysmaster** database does not already exist, the database server keeps trying to create it each time that you start the database server.



Warning: The logical logs continue to fill with the transactions that result from the creation of the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, the database server halts with a long-transaction error. Thus, you must back up the logical logs.

Execute the **ontape -a** or **ontape -c** command. Once the logical logs are backed up, processing resumes.

When the system reaches quiescent mode successfully, you can note whether you need to adjust the shared memory and semaphore values for your operating system, which you configured in [“Reconfigure the UNIX Operating System” on page 12-31](#). Check your database server message log for status messages that pertain to the change to quiescent mode.

To bring the target database server from quiescent to on-line mode

To change the database server mode from quiescent to on-line, execute the following command:

```
onmode -m
```

At this point, the system attempts to rebuild system catalog indexes. If you try to access them at this time, you might find some of them locked. If these catalogs cannot be rebuilt, you might need to allocate more space in your root dbspace. Along with the 1,100 extra pages that you need to create the **sysmaster** database, you must add the number of additional pages that the formula gives in [“Accommodating System Catalog Indexes” on page 12-13](#). Return to this section and allocate sufficient space.

Upgrade User-Table Indexes

If you have not already done so, use the formula provided in [“Accommodating User-Table Indexes” on page 12-16](#) to allocate the additional disk space that you need for user-table indexes in your target database server. You can now begin upgrading user-table indexes.

Upgrading user-table indexes requires planning. You can use several methods to upgrade indexes for various tables. Depending on the size and demand for a table, you might prefer one method over another. If you have not already done so, generate a list of tables in each of your databases and identify the appropriate upgrade method for each index, as described in [“Accommodating the Upgrading of User-Table Indexes” on page 12-18](#).

Before you make the database server available to regular users, execute the **oncheck -cI -y** command to upgrade indexes for large or critical tables, as described in [“Use the oncheck Utility to Upgrade Indexes” on page 12-35](#). You can defer smaller and less-critical tables until after the database server is brought on-line. You can then upgrade these tables through SQL with the UPDATE STATISTICS statement or the DROP INDEX and CREATE INDEX statements.

You might be able to expedite the process of upgrading indexes by taking advantage of enhanced features that are provided with your target database server such as support for multiple CPUs, parallel index builds, and so on. For more information, refer to your *Administrator’s Guide*.

The new indexes will be larger than the old indexes no matter which method you choose. Make sure that you allocate enough room in each dbspace for the new indexes. Otherwise, you might need to move tables to other dbspaces before you upgrade indexes, as described in the following section, “[Move Tables to Another dbspace.](#)”

Move Tables to Another dbspace

If you neglected to make the proper adjustments to your dbspaces as described under “[Accommodating User-Table Indexes](#)” on page 12-16, you might need to move one or more tables to another dbspace to accommodate the larger index in your target database server. If you know that you have enough room for your new indexes, you can skip ahead to “[Use the oncheck Utility to Upgrade Indexes.](#)”

You can use either of the following techniques to move tables during the upgrade process:

- The INSERT statement
- The **onunload** and **onload** utilities

To use INSERT to move tables to a new dbspace

1. Create a table in the new dbspace with a temporary name.
2. Use `INSERT INTO newtab SELECT * FROM oldtab.`
3. Drop the old table.
4. Create all the indexes in the new table.
5. Rename the new table to the original name.

To use onunload and onload to move tables to a new dbspace

1. Create a new table in another dbspace.
2. Drop all the indexes from the old table.
3. Use **onunload** to copy the old table to a file.
4. Drop the table and then update the database to create a new version of the table in the new dbspace.
5. Use **onload** to load the data back into the table.
6. Create all the indexes in the new table.

Use the oncheck Utility to Upgrade Indexes

The **oncheck** utility can be helpful when you upgrade ANSI-compliant databases, and in situations where critical tables must be made accessible as soon as possible.

You can run **oncheck -cI -y** to upgrade indexes for a single table or an entire database when your target database server is in quiescent mode. After you make sure that no users are present, use **onmode -s** to bring the system into quiescent mode.

To upgrade indexes for a single table

```
oncheck -cI -y database_name:table_name
```

database_name is the name of the database.

table_name is the name of the table.

To upgrade indexes for the entire database

```
oncheck -cI -y database_name
```

The **oncheck** command displays messages of the following form:

```
Index index-name is bad. OK to repair it?
```

This message indicates that the existing (4.1 or 5.0) index is not in the correct format for your target database server. The **-y** option automatically answers **yes** to this prompt, allowing your target database server to upgrade each index in turn automatically.

Use UPDATE STATISTICS to Upgrade Indexes



Tip: The **UPDATE STATISTICS** statement includes new options that were introduced in OnLine Dynamic Server 6.0. For details, refer to the “Informix Guide to SQL: Syntax.”

You can execute an **UPDATE STATISTICS** statement to upgrade the indexes for a single table or, in some circumstances, for an entire database. The target database server **UPDATE STATISTICS** statement automatically upgrades Version 4.1 and 5.0 indexes, provided that the dbspace has enough space. For more information, see [“Move Tables to Another dbspace” on page 12-34](#).



This method is especially useful for nonlogging databases. Or you can use UPDATE STATISTICS in a database without ANSI logging by issuing the statement *outside of a transaction*. When you take the proper precautions, you can even use this method within a transaction or an ANSI-compliant database.

Warning: *If you execute an UPDATE STATISTICS statement within a transaction, you must follow up with a COMMIT WORK or ROLLBACK WORK statement to close the transaction. Otherwise, the Dynamic Server 7.3 or OnLine Dynamic Server instance eventually halts with a long-transaction error.*

When you upgrade indexes within a transaction or an ANSI-compliant database, Informix recommends that you use one of the following approaches to limit the risk of encountering a long transaction:

- Limit the scope of each UPDATE STATISTICS statement to a single table. This approach reduces your risk of encountering a long transaction. Execute separate UPDATE STATISTICS statements for each table, followed by separate COMMIT WORK statements.
- Update statistics for an entire database only when no other users have access to the current instance of your target database server.
- Use **oncheck** to upgrade indexes while the database server is in quiescent mode. For more information, see [“Use the oncheck Utility to Upgrade Indexes” on page 12-35](#).

Use DROP INDEX and CREATE INDEX to Upgrade Indexes

If neither of the previous two methods seem suitable, you can drop and rebuild the indexes for your tables individually. When you drop and rebuild indexes, you can override the default fill factor that is specified in the FILLFACTOR parameter of the ONCONFIG file. For details, refer to the *Informix Guide to SQL: Syntax*.

Verify the Integrity of the Data

Informix recommends that you verify the integrity of your data before you run the reversion utility. [Figure 12-4](#) lists the **oncheck** commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 12-4
Commands for
Verifying Data
Integrity



Tip: You might see the following warning. It means that you have not defined any synonyms.

```
WARNING: No syssyntax records found.
```

Make an Initial Backup of the Target Database Server

Use your target database server backup tool (ON-Bar, ON-Archive, or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made your final backup of your database server. For more information, refer to your *Archive and Backup Guide* or your *Backup and Restore Guide*.



Important: Do not restore the backed up logical logs to the newer version of your database server from your source database server.

Add rowid Columns to Fragmented Tables

In the Version 7.x database servers, fragmented tables do not have a **rowid** field by default. If you migrate fragmented data tables to Dynamic Server 7.3 or OnLine Dynamic Server 7.x, you must add a **rowid** column to each table before you use applications that use **rowid** values to access data.

Complete Migration

When you finish the level-0 backup, the migration process is complete and users can use your database server to access data safely.

The first time you bring up the target database server, the **sysmaster** database is built in 15 to 30 minutes. Check the message log to ensure that the **sysmaster** database build has completed before you allow users to access the database server.

After you successfully migrate to your target database server, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between your earlier version and your target database server. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on topics related to performance, refer to your *Performance Guide*.

GLS

Convert a Database to a Different Locale

When you complete the migration to Dynamic Server 7.3 or OnLine Dynamic Server from OnLine 4.1 or 5.0, your databases use the default locale, U.S. English. When the target database server is running successfully, you can take the following steps to convert a database to a non-default locale.

To convert a database to a non-default locale

1. Use the UNLOAD statement or the **dbexport** utility to unload data from the database to be converted.
2. Set the **CLIENT_LOCALE** and **DB_LOCALE** environment variables to support the new GLS locale.
3. Create a database with the new locale by issuing the CREATE DATABASE statement from an application that has the correct client locale variables set in its environment.
4. Modify the schema representation to replace CHAR and VARCHAR columns with NCHAR and NVARCHAR columns, respectively.
5. Use the LOAD statement or the **dbimport** utility to load the data into the new database.

For more information about the GLS feature, refer to Section VI, “[Locale Changes](#),” and to the *Informix Guide to GLS Functionality*.

Reverting to OnLine 5.0 or 4.1

This section describes the process to revert to OnLine, Version 5.0 or Version 4.1, from Dynamic Server 7.3 or OnLine Dynamic Server. These steps also apply if you are reverting from Version 7.2x, 7.10.UD1, 7.10.UCx, or 6.0. Before you can revert to OnLine 5.0 or OnLine 4.1 from Dynamic Server 7.3 or OnLine Dynamic Server, you must modify the configuration limits and remove constructs that OnLine 5.0 or OnLine 4.1 does not support.

The procedure for reverting to OnLine includes the following steps, which the rest of this section describes:

1. Save copies of the current configuration files.
2. Remove database server users.
3. Verify the integrity of the data.
4. Back up the source database server.
5. Remove features that later versions introduced.
6. Run the reversion utility (**onmode -b**).
7. Prepare the **TBCONFIG** configuration file.
8. Reset environment variables.
9. Bring up OnLine 5.0 or 4.1 in quiescent mode.
10. Verify the integrity of the data.
11. Back up OnLine 5.0 or 4.1.
12. Return OnLine 5.0 or 4.1 to on-line mode.

Save Copies of the Current Configuration Files

Before you start the reversion process, save copies of your configuration files.

Remove Database Server Users

Remove all users from the source database server before you begin the reversion process. Warn the users that you plan to shut down the database server and then execute the following command:

```
onmode -s
```

The **-s** flag on **onmode** restricts new access to the database server but allows current processing to finish. When all processing is finished, the database server goes to quiescent mode, and you can continue the reversion process.

Verify the Integrity of the Data

Before you allow users to access the databases, use the **oncheck** utility to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. For more information, see [“Verify the Integrity of the Data” on page 12-37](#).

Back Up the Source Database Server

Informix recommends that you use **ontape** or ON-Archive to make a level-0 backup. For details on how to make backups, refer to your *Archive and Backup Guide*.

Remove Features That Later Versions Introduced

Before you can revert to OnLine 5.0 or 4.1, you must remove features that later versions of the database server introduced, as follows:

- Remove NLS and GLS databases.
- Disable data replication.
- Change all fragmented tables back into unfragmented tables. See your *Administrator's Guide* for instructions on how to defragment tables.
- Disable role separation.
- Remove unsupported SQL statements.
- Adjust configuration parameters.

Figure 12-5 and Figure 12-6 show configuration parameter limits for Version 5.0 and Version 4.1.

Figure 12-5
Configuration Parameter Limits for Version 5.0

Resource	Parameter	5.0 Limit
Logical logs	LOGFILES, LOGSMAX	<= 1 page of entries
Dbspaces	DBSPACES	<= 40
Chunks	CHUNKS	<= 1 page of entries
Buffers	BUFFERS	<= 512 kilobytes

Figure 12-6
Configuration Parameter Limits for Version 4.1

Resource	Parameter	4.1 Limit
Logical logs	LOGFILES, LOGSMAX	<= 1 page of entries
Dbspaces	DBSPACES	<= 40
Chunks	CHUNKS	<= 1 page of entries
Buffers	BUFFERS	<= 32,000

The chunk limit depends on the length of the pathname that you choose for the chunk. The limit could range from 13 to 58.

Tip: Logical-log files of differing sizes that were created after 6.0 initialization do not impair the reversion.



Run the Reversion Utility

To restore the database server files to a form that is compatible with OnLine 5.0, execute the following command:

```
onmode -b 5.0
```

The reversion utility includes an implicit **onmode -yuk**. This command forcibly removes all users. After the reversion is complete, the system is in off-line mode.



Tip: The **onmode -b** command also rebuilds the user-table indexes automatically.

Depending on the version of the database server from which you are reverting, the **onmode** utility performs the following actions:

- Verifies that no GLS databases exist.
- Verifies that data replication is off.
- Removes the second slot in the archive reserved page for data replication.
- Drops the **sysmaster** database.
- Rewrites leaf pages of all indexes.
- Frees reserved-page extensions.
- Removes the data-replication slot from the archive reserved page.
- Rewrites all **partnums** on disk (systables, database tablespace, tablespace pages, blob freemap pages) in their old formats.
- Rewrites dbspace page in the old format.
- Writes a checkpoint record in Version 5.0 format to a clean logical-log file.

The reformatting does not make the data space identical to its earlier format. Some of the changes made during an upgrade to Version 7.3 from an earlier version do not make the space incompatible with earlier versions, and the **-b** option does not modify these changes.

Prepare the TBCONFIG Configuration File

Modify the configuration file to eliminate parameters that OnLine 5.0 or 4.1 does not recognize. You might find it easier to compare your saved configuration file with the **tbconfig.std** file and make adjustments accordingly.

Be sure to include the **USERS** configuration parameter, which was replaced by **USERTHREADS** in Version 6.0 and later.



Important: Use the same values for **ROOTPATH**, **ROOTSIZE**, and **ROOTOFFSET** for both versions of your database server.

Reset Environment Variables

Reset the environment variables to values that are appropriate for the selected version of your database server. Remember that in Version 5.0 and Version 4.1 you specify the configuration file with the **TBCONFIG** environment variable instead of **ONCONFIG**.

The **PDQPRIORITY** environment variable and the SQL **SETPDQPRIORITY** statement were introduced after Version 6.0. If you set the **PDQPRIORITY** environment variable, it does not cause problems but it might cause confusion. You must remove the **SETPDQPRIORITY** statement from your applications.

Bring Up OnLine 5.0 or 4.1 in Quiescent Mode

Bring up OnLine 5.0 or 4.1 from off-line to quiescent mode. Execute the following command:

```
tbinit -s
```

Verify the Integrity of the Data

Before you allow users to access the databases, check the integrity of the data. [Figure 12-7](#) lists the **tbcheck** commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	tbcheck -cr
Check system catalog tables	tbcheck -cc <i>database_name</i>
Check data	tbcheck -cD <i>database_name</i>
Check indexes	tbcheck -cI <i>database_name</i>

Figure 12-7
Commands for
Verifying the Data
Integrity

When you run **tbcheck**, you might see the following message:

```
OLD pn_bytes != NEW pn_nbytes
```

This message does not require any action on your part. It indicates that a later version of the database server accessed your database.

Back Up OnLine 5.0 or 4.1

After you complete the reversion, Informix recommends that you make a level-0 backup. Use the **tbtape** utility to prepare backups. For details on making backups, refer to the appropriate *Administrator's Guide*.

Return Online 5.0 or 4.1 to On-Line Mode

To change your OnLine 5.0 or 4.1 mode from quiescent to on-line, execute the following command:

```
tbmode -m
```

The reversion is now complete, and users can access the data.

Migrating Between INFORMIX-SE Database Servers and Converting C-ISAM Files

In This Chapter	13-3
Importing and Exporting Data	13-3
Migrating Between Different Versions of SE	13-4
Preparing to Upgrade to SE 7.2 from SE 5.0 or 4.1	13-4
The sqlhosts File	13-4
Environment Variables	13-4
SE Utilities	13-5
Preparing to Upgrade to SE 7.2 from SE 7.1 or 6.0	13-5
Reverting to an Earlier Version of SE.	13-6
Converting C-ISAM Files to SE.	13-6

In This Chapter

This chapter describes how to migrate between different versions of INFORMIX-SE database servers. The latest version of SE is Version 7.2x. [Chapter 14, “Migrating Between 7.x Database Server Options and Editions,”](#) covers moving data between SE and Dynamic Server 7.3.

Importing and Exporting Data

You can use the following utilities to import data to and export data from an SE database server:

- **dbexport/dbimport**
- **UNLOAD/dbschema/LOAD**
- **UNLOAD/dbschema/dbload**

Let your target destination, performance, and ease-of-use requirements determine the method that you use. Refer to [Figure 2-2, “Comparison of Tools for Loading Data” on page 2-8](#) for details. [Chapter 14, “Migrating Between 7.x Database Server Options and Editions,”](#) describes the procedures for exporting and importing data. Section VIII, [“Data Migration Utilities,”](#) describes the syntax of the **dbexport**, **dbimport**, **UNLOAD**, **dbschema**, **LOAD**, and **dbload** utilities.

Migrating Between Different Versions of SE

Unlike Dynamic Server 7.3 data, SE data is stored in ordinary UNIX files. The structure of these files remained the same between versions of SE, so that migrating from one version to a later version requires little preparation. SE manages the file contents, but the operating system manages the I/O.

Preparing to Upgrade to SE 7.2 from SE 5.0 or 4.1

Version 6.0 of Informix products introduced changes in the way clients connect to database servers. The names of database server utilities also changed.

The sqlhosts File

Pre-6.0 versions of SE did not require an **sqlhosts** file on UNIX unless you used SE with INFORMIX-NET. Beginning with Version 6.0, Informix products require an **sqlhosts** file to specify connections between clients and servers. For information about how to prepare your **sqlhosts** file, refer to your *Administrator's Guide*.

Environment Variables

Beginning with Version 6.0, SE requires the following environment variables:

- INFORMIXDIR
- PATH
- INFORMIXSERVER

Depending on your network configuration, you might also need the following environment variables:

- SQLEXEC
- SQLRMDIR
- SQLRM

For information about how to set these environment variables, refer to your *Administrator's Guide*.

SE Utilities

Beginning with Version 6.0, the names of the SE utilities **bcheck** and **dblog** changed to **secheck** and **selog**, respectively. If you have scripts that use these utilities, you must update the names of the utilities.

To migrate to SE 7.2x from a pre-6.0 version of SE

1. Ask all users to exit from their applications.
2. Verify the validity of your data.

For Version 4.1 or 5.x, execute the following command for each table in the database:

```
bcheck tablename
```

For more information about these utilities, refer to the appropriate version of your *Administrator's Guide*.

3. Install SE, Version 7.2x.

For information about how to install SE, refer to your *Installation Guide*.

Preparing to Upgrade to SE 7.2 from SE 7.1 or 6.0

When you migrate to SE 7.2x from a post-6.0 version of SE, you do not need to change the environment variables or the **sqlhosts** file.

To migrate to SE 7.2x from a post-6.0 version of SE

1. Ask all users to exit from their applications.
2. Verify the validity of the data.

For Version 6.0 and later, execute the following command for each table in the database:

```
secheck tablename
```

For more information about these utilities, refer to the appropriate version of the *INFORMIX-SE Administrator's Guide*.

3. Install SE, Version 7.2x.

For information about how to install SE, refer to your *Installation Guide*.

Reverting to an Earlier Version of SE

Informix does not provide a reversion utility for SE. To move your database to an earlier version of SE, you must unload and then reload your data. For information about how to use **dbexport**, **dbimport**, and UNLOAD to move your data, see [“Migrating Between Different Versions of SE” on page 13-4](#).

Converting C-ISAM Files to SE

This section describes how to convert C-ISAM files to SE format. C-ISAM files are organized differently than relational tables. C-ISAM files tend to be much larger and are not normalized. Therefore, you need to convert C-ISAM applications to the SQL format that you can use with SE, as follows:

1. In SE, use the SQL CREATE TABLE statement to create a table that corresponds to the data fields in the C-ISAM application. (Do not use the name of the C-ISAM file for the table name.)
2. Delete the empty **.dat** file that SE created in the CREATE TABLE statement. It has the name `tablename|tabid|.dat`. Also, delete the empty **.idx** file.
3. Either move the C-ISAM file to the **.dbs** directory or update the **systables** system catalog with the name of the C-ISAM data file.
If you update both the C-ISAM and SE files with logging turned on, you must use common logging.

Migrating Between 7.x Database Server Options and Editions

In This Chapter	14-5
Moving Data Between Database Servers	14-6
Using UNLOAD, dbschema, and LOAD	14-6
Using UNLOAD, dbschema, and dbload	14-8
Migrating to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3	14-8
Save a Copy of the Current Configuration Files.	14-9
Verify the Integrity of the Data	14-10
Back Up Dynamic Server 7.3	14-10
Shut Down Dynamic Server 7.3	14-10
Install and Configure Dynamic Server, Workgroup and Developer Editions	14-12
Verify Port Numbers and Services File.	14-12
Customize the Environment for Dynamic Server, Workgroup and Developer Editions	14-12
Bring Dynamic Server, Workgroup and Developer Editions, On-Line.	14-13
Verify the Integrity of the Data	14-13
Back Up Dynamic Server, Workgroup and Developer Editions	14-13
Update Statistics	14-14
Complete Migration	14-14
Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions	14-14

Migrating to Dynamic Server 7.3 from Dynamic Server, W/D	14-15
Save a Copy of the Current Configuration Files	14-16
Verify the Integrity of the Data	14-16
Back Up Dynamic Server, Workgroup and Developer Editions	14-17
Shut Down Dynamic Server, Workgroup and Developer Editions	14-17
Install and Configure Dynamic Server 7.3	14-18
Verify Port Numbers and Services File	14-19
Customize the Environment for Dynamic Server 7.3	14-19
Bring Dynamic Server 7.3 On-Line	14-19
Verify the Integrity of the Data	14-20
Back Up Dynamic Server 7.3	14-20
Update Statistics	14-20
Complete Migration	14-20
Adapt Your Programs for Dynamic Server 7.3	14-21
Migrating Between Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server	14-21
Using Different Server and Client Configurations	14-22
Workgroup Edition 7.3 and OnLine Workgroup Server 7.22 through 7.24 and IECC 3.0	14-22
OnLine Workgroup Server 7.12 and IECC 2.0	14-22
OnLine Workgroup Server 7.22 and IECC 2.0	14-22
OnLine Workgroup Server 7.12 and IECC 1.0	14-23
Preparing for Migration	14-23
Check Available Space and System Requirements.	14-24
Save a Copy of the Current Configuration Files	14-24
Close All Transactions and Place the Source Database Server in Quiescent Mode	14-25
Verify the Integrity of the Data	14-26
Back Up the Source Database Server	14-27
Upgrading to Dynamic Server, Workgroup and Developer Editions, Version 7.3	14-27
Save an Output File of SQL Statements for Access Paths	14-28
Bring the Source Database Server Off-Line	14-28
Change UNIX Kernel Parameters	14-29
Install the Target Database Server and Administration Tools	14-29
Customize the Database Server Environment	14-31

Prepare the Target Database Server to Use Enterprise Replication	14-32
Install and Configure SNMP	14-33
Bring the Target Database Server On-Line	14-34
Verify the Integrity of the Data	14-35
Make an Initial Complete Backup of the Target Database Server	14-35
Run UPDATE STATISTICS	14-36
Verify the Access Paths of Your SQL Statements	14-36
Complete Migration	14-37
Changing Database Server Definitions on Windows NT	14-37
Migrating to a GLS Locale.	14-38
Reverting the Database Server and IECC	14-38
Remove Unsupported SQL Features	14-39
Uninstall Enterprise Replication or Workgroup Replication, Version 7.3	14-39
Run the Reversion Utility	14-40
Uninstall the Database Server and IECC	14-40
Remove GLS Features (Only If Reverting to 7.12)	14-41
Modify Configuration Parameters	14-41
Reset Environment Variables	14-42
Reinstall the Old Version of the Database Server.	14-43
Bring the Target Database Server On-Line	14-44
Verify the Integrity of the Data	14-45
Make an Initial Complete Backup of the Target Database Server	14-45
Complete Reversion	14-45
Moving Data Between SE and Dynamic Server 7.3 or Its Editions.	14-45
Moving Data to Dynamic Server 7.3 or Its Editions from SE	14-45
Using the dbexport and dbimport Utilities	14-46
Using the UNLOAD Statement	14-47
Moving C-ISAM Data to Dynamic Server or Its Editions	14-49
Adapting Programs for Dynamic Server 7.3 or Its Editions	14-49
Moving Data to SE from Dynamic Server or Its Editions	14-50
Eliminating Features That SE Does Not Support.	14-50
Using the dbexport and dbimport Utilities	14-50
Using the UNLOAD Statement	14-52
Adapting Your Programs for SE	14-53

In This Chapter

This chapter describes the steps for how to migrate between Version 7.x database servers on the same operating system. [Figure 1-2 on page 1-6](#) and [Figure 1-3 on page 1-7](#) show the paths for migrating between different database servers in the same environment.

This chapter covers the following topics:

- Moving data between database servers
- Migrating to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3
- Migrating to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions
- Migrating between Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server
- Moving data between SE and Dynamic Server



Tip: For information on how to move the database server between UNIX and Windows NT, refer to [Chapter 15, “Moving a Database Server to a Different Operating System.”](#)

Moving Data Between Database Servers

The method that you choose for moving data depends on how much data you plan to move. All these methods deliver similar performance and allow you to modify the schema of the database. You can use the following migration methods to move data between database servers:

- **dbexport** and **dbimport**
To move an entire database, use the **dbexport** and **dbimport** utilities.
- **UNLOAD**, **LOAD**, and **dbschema**
To move selected columns or tables, use the **UNLOAD** statement. Use **LOAD** when you do not want to change the data format.
- **UNLOAD**, **dbload**, and **dbschema**
To move selected columns or tables, use the **UNLOAD** statement. Use **dbload** to change the data format.
- **onunload** and **onload**
To move data in page-sized chunks, use the **onunload** utility. Use the **onunload** utility to move data to an identical database server on a computer of the same type.
- The High-Performance Loader (HPL)
To move selected columns or tables or an entire database, use the HPL.

SE stores data in a different format than does Dynamic Server 7.3. When your migration involves SE, you must export data and its schema information from one database server and import the exported data into the other database server.

Using **UNLOAD**, **dbschema**, and **LOAD**

The **UNLOAD** statement lets you retrieve selected rows from a database and write those rows to a text file. If you want to move selected tables or columns instead of an entire database between Dynamic Server 7.3 and Dynamic Server, Workgroup and Developer Editions, use the **UNLOAD** and **LOAD** statements in the DB-Access utility with the **dbschema** utility.

To use UNLOAD, dbschema, and LOAD to move data to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. To move the selected data into text files, use the UNLOAD statement. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the source database server.
 - b. Edit the schema file so that it describes the new table or tables.
If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.
6. Follow the instructions in your *Installation Guide* and your *Administrator's Guide* to install and configure the target database servers.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.
10. Select the target database.
11. If you are creating a new database, execute the CREATE DATABASE statement or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table or tables, choose and run the schema file that you prepared in step 5, or enter CREATE TABLE statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. To load the data into the desired tables, use LOAD statements.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data in the specified UNLOAD file before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move data between Dynamic Server and Dynamic Server, Workgroup and Developer Editions

1. Follow steps 1 through 13 from [“Using UNLOAD, dbschema, and LOAD” on page 14-6.](#)
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to Section VIII, [“Data Migration Utilities.”](#) For information on how to use DB-Access, refer to the *DB-Access User Manual*.

Migrating to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3

This section describes the steps for migrating to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3 on the same operating system.

Migration to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3 is automatic if the database servers share the same operating system. Automatic migration means you do not need to use a migration utility such as **dbexport** to move the data; the migration occurs automatically when you bring up the target database server. The same operating system can be UNIX on the same computer, Windows NT on the same computer, or the same version of Windows NT on different computers.

For information about migrating to a different operating system, see [“Moving Data to a Database Server on a Different Operating System” on page 15-5.](#)

To migrate on the same operating system, complete the following migration steps, which later sections describe in detail:

1. Save a copy of the current configuration files.
2. Verify the integrity of the data.
3. Make a final complete backup of your source database server.
4. Bring the source database server off-line.
5. Install and configure the target database server.
6. Verify port numbers and **services** file.
7. Customize the database server environment. (This step is optional.)
8. Bring the target database server on-line.
9. Verify the integrity of the data.
10. Make an initial, complete backup of the target database server.
11. Run UPDATE STATISTICS.
12. Complete migration.
13. Adapt your programs for Dynamic Server, Workgroup and Developer Editions.

Save a Copy of the Current Configuration Files

Save a copy of the current configuration files that you have modified. These should include the following files:

- Current **ONCONFIG** file, located in the **etc** subdirectory of your installation directory
- **sqlhosts** information
- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory
- ON-Archive configuration files, located in the **etc** subdirectory

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data. [Figure 14-1](#) lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	tbcheck -cc <i>database_name</i>
Check data	tbcheck -cD <i>database_name</i>
Check indexes	tbcheck -cI <i>database_name</i>

Figure 14-1
Commands for
Verifying Data
Integrity

For information about **oncheck**, refer to your *Administrator's Guide*.

Back Up Dynamic Server 7.3

Use your preferred backup method to make a complete (level-0) backup of Dynamic Server 7.3.

The Windows NT environment does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. For more information on how to make backups, refer to your *Archive and Backup Guide*.

Shut Down Dynamic Server 7.3

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

WIN NT

WIN NT

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX. On Windows NT, you must be a member of the **Informix-Admin** group.

Use **Services** in the Windows NT **Control Panel**. ♦

3. Execute the following command to take the database server to quiescent mode:

```
onmode -sy
```

4. Wait until your database server is in quiescent mode.

To verify the mode of your database server, execute the **onstat** - command. The first line of the **onstat** output contains the status of your database server. [Figure 14-2](#) shows that the database server is in quiescent mode.

```
Informix Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx -- xxxx Kbytes
```

Informix Dynamic Server is in quiescent mode.

Figure 14-2
Example of **onstat**
Status Line

5. Execute the following command to force a new logical log:

```
onmode -l
```
6. Execute the following command to force a checkpoint:

```
onmode -c
```
7. Execute the following command to shut down the database server:

```
onmode -yuk
```

Tip: Monitor your log activity to verify that all commands were executed properly and to check for inconsistencies prior to migration.



Install and Configure Dynamic Server, Workgroup and Developer Editions

If you have not already done so, follow the instructions in your *Installation Guide* to install and configure your target database server.

You can install Dynamic Server, Workgroup and Developer Editions, and Informix Enterprise Command Center (IECC) either on the same computer or on different computers. You can install IECC on a personal computer that runs Windows 95 or Windows NT and the database server on a UNIX computer. The installation program also starts the server agent, which is the communications link between Dynamic Server, Workgroup and Developer Editions, and the IECC client.

Use the **Setup** program to specify the network protocol and the computer on which Dynamic Server, Workgroup and Developer Editions, looks for the database server definitions (**sqlhosts** and **osahosts** definitions).

Verify Port Numbers and Services File

The **services** file contains service names, port numbers, and protocol information. If you installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and on the server computers.

For UNIX operating systems not running NIS, the **services** file resides in the **/etc/services** directory on the database server and in the **\windir\services** directory on the Windows 95 client. ♦

The **services** file resides in the **\windir\system32\drivers\etc** directory. ♦

Customize the Environment for Dynamic Server, Workgroup and Developer Editions

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for Dynamic Server, Workgroup and Developer Editions. Use a text editor to edit the ONCONFIG file. For more information on configuration parameters, refer to your *Installation Guide*.

Important: Use the same values for your target database server for **ROOTOFFSET**, **ROOTSIZE**, and **ROOTPATH** that you used for your source database server.

UNIX

WIN NT



You might want to customize new environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Bring Dynamic Server, Workgroup and Developer Editions, On-Line

UNIX

To start your database server at the UNIX command line, enter `oninit` on the server computer. ♦

WIN NT

The installation program on Windows NT brings the target database server on-line automatically. ♦

If you customized the database server environment, you can bring down and restart your database server with IECC. For more information, refer to the *Informix Enterprise Command Center User Guide*.

When you restart Dynamic Server, Workgroup and Developer Editions, the changes to the configuration parameters and environment variables take effect.

Verify the Integrity of the Data

To verify the integrity of data, use the `oncheck` utility, as [“Verify the Integrity of the Data” on page 14-10](#) describes.

Back Up Dynamic Server, Workgroup and Developer Editions

Make a complete, whole-system backup of your target database server.

UNIX

On UNIX, use ON-Bar, **ontape**, or ON-Archive to make the backup. ♦

WIN NT

On Windows NT, double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group. ♦

For more information on backing up, refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.

Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement on the database server. The UPDATE STATISTICS statement updates the information that Dynamic Server, Workgroup and Developer Editions, uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Complete Migration

The first time your target database server is brought on-line, the **sysmaster** and the **sysutils** databases are built. Check the message log to ensure that the **sysutils** and **sysmaster** databases are created successfully before you allow users to access the database server. After you ensure that client users can access data on your target database server, the migration process is complete.

Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions

After you successfully migrate the database server data, verify that your application developers know the differences between the source and target database servers. Dynamic Server, Workgroup and Developer Editions, supports the same features as Dynamic Server 7.3, except for the High-Performance Loader (HPL) and the following features:

- Fragmentation (also known as partitioning)
- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that Dynamic Server, Workgroup and Developer Editions, supports, refer to the *Informix Guide to SQL: Syntax* and the *Informix Guide to SQL: Reference*.

Migrating to Dynamic Server 7.3 from Dynamic Server, W/D

This section describes the steps for migrating to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on the same operating system.

Migration to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, is automatic if the database servers share the same operating system. Automatic migration means you do not need to use a migration utility such as **dbexport** to move the data; the migration occurs automatically when you bring up the target database server. The same operating system can be UNIX on the same computer, Windows NT on the same computer, or the same version of Windows NT on different computers.

For information about migrating to a different operating system, see [“Moving Data to a Database Server on a Different Operating System” on page 15-5.](#)

To migrate on the same operating system, complete the following migration steps, which later sections describe in detail:

1. Save a copy of the current configuration files.
2. Verify the integrity of the data.
3. Make a final complete backup of your source database server.
4. Bring your source database server off-line.
5. Install and configure the target database server.
6. Verify port numbers and **services** file.
7. Customize the database server environment. (This step is optional.)
8. Bring the target database server on-line.
9. Verify the integrity of the data.
10. Make an initial, complete backup of the target database server.
11. Run UPDATE STATISTICS.
12. Complete migration.
13. Adapt your programs for Dynamic Server 7.3.



Important: If you want to install on Windows NT, select the **Copy all files, but leave configuration alone** option on the **Run Installation Again** page. The installation program automatically copies the new database server files and saves the configuration and the database data.

Do not select the **Copy all files and reconfigure the product** option, or your configuration and database information will be lost.

After you ensure that client users can access data on the target database server, the migration process is complete.

Save a Copy of the Current Configuration Files

Save a copy of the current configuration files that you have modified. These should include the following files:

- Current **ONCONFIG** file, located in the **etc** subdirectory of your installation directory
- **sqlhosts** information
- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory
- ON-Archive configuration files, located in the **etc** subdirectory

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data. [Figure 14-3](#) lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	tbcheck -cc database_name
Check data	tbcheck -cD database_name
Check indexes	tbcheck -cI database_name

Figure 14-3
Commands for
Verifying Data
Integrity

For information on **oncheck**, refer to your *Administrator's Guide*.

Back Up Dynamic Server, Workgroup and Developer Editions

Use your preferred backup method to make a complete (level-0) backup of Dynamic Server, Workgroup and Developer Editions.

The Windows NT environment does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup. For more information on how to make backups, refer to your *Archive and Backup Guide*.

Shut Down Dynamic Server, Workgroup and Developer Editions

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX. On Windows NT, you must be a member of the **Informix-Admin** group.

Use **Services** in the Windows NT **Control Panel**. ♦

3. Execute the following command to take the database server to quiescent mode:

```
onmode -sy
```

4. Wait until your database server is in quiescent mode.

To verify the mode of your database server, execute the **onstat -** command. The first line of the **onstat** output contains the status of your database server. [Figure 14-4](#) shows that the database server is in quiescent mode.

WIN NT

WIN NT

```
Informix Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx -- xxxx Kbytes
```

```
Informix Dynamic Server is in quiescent mode.
```

Figure 14-4
*Example of onstat
Status Line*



5. Execute the following command to force a new logical log:

```
onmode -l
```

6. Execute the following command to force a checkpoint:

```
onmode -c
```

7. Execute the following command to shut down the database server:

```
onmode -yuk
```

Tip: Monitor your log activity to verify that all commands were executed properly and to check for inconsistencies prior to migration.

Install and Configure Dynamic Server 7.3

If you have not already done so, follow the instructions in your *Installation Guide* to install and configure your target database server.

You can install Dynamic Server 7.3 and IECC either on the same computer or on different computers. You can install IECC on a personal computer that runs Windows 95 or Windows NT and the database server on a UNIX computer. The installation program also starts the server agent, which is the communications link between Dynamic Server 7.3 and the IECC client.

Use the **Setup** program to specify the network protocol and the computer on which Dynamic Server 7.3 looks for the database server definitions (**sqlhosts** and **osahosts** definitions).

Verify Port Numbers and Services File

The **services** file contains service names, port numbers, and protocol information. If you installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and on the server computers.

UNIX

For UNIX operating systems not running NIS, the **services** file resides in the **/etc/services** directory on the database server and in the **\windir\services** directory on the Windows 95 client. ♦

WIN NT

The **services** file resides in the **\windir\system32\drivers\etc** directory. ♦

Customize the Environment for Dynamic Server 7.3

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for Dynamic Server 7.3. Use a text editor to edit the ONCONFIG file. For more information on configuration parameters, refer to your *Installation Guide*.



Important: Use the same values for your target database server for **ROOTOFFSET**, **ROOTSIZE**, and **ROOTPATH** that you used for your source database server.

You might want to customize new environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Bring Dynamic Server 7.3 On-Line

UNIX

To start your database server at the UNIX command line, enter **oninit** on the server computer. ♦

WIN NT

The installation program on Windows NT brings the target database server on-line automatically. ♦

If you customized the database server environment, you can bring down and restart your database server with IECC. For more information, refer to the *Informix Enterprise Command Center User Guide*.

When you restart Dynamic Server 7.3, the changes to the configuration parameters and environment variables take effect.

UNIX

WIN NT

Verify the Integrity of the Data

To verify the integrity of data, use the **oncheck** utility, as [“Verify the Integrity of the Data” on page 14-10](#) describes.

Back Up Dynamic Server 7.3

Make a complete, whole-system backup of your target database server.

On UNIX, use ON-Bar, **ontape**, or ON-Archive to make the backup. ♦

On Windows NT, double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group. ♦

For more information on backing up, refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.

Update Statistics

After you complete the migration procedure, run the UPDATE STATISTICS statement on the server. The UPDATE STATISTICS statement updates the information that Dynamic Server 7.3 uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Complete Migration

The first time the target database server is brought on-line, the **sysmaster** and **sysutils** databases are built. Check the message log to ensure that the database builds are complete before you allow users to access the database server. After you ensure that client users can access data on your target database server, the migration process is complete.

Adapt Your Programs for Dynamic Server 7.3

After you successfully migrate to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3, verify that your application developers know the differences between both database servers. Dynamic Server, Workgroup and Developer Editions, supports the same features as Dynamic Server 7.3 except for the High-Performance Loader (HPL) and the following features:

- Fragmentation (also known as partitioning)
- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that Dynamic Server, Workgroup and Developer Editions, supports, refer to the *Informix Guide to SQL: Syntax* manual and the *Informix Guide to SQL: Reference*.

Migrating Between Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server

This section covers the following topics:

- Using different server and client configurations
- Preparing for migration
- Upgrading to Dynamic Server, Workgroup and Developer Editions, Version 7.3
- Reverting the database server and IECC

UNIX

WIN NT

Using Different Server and Client Configurations

This section discusses the effect of different database server and client configurations. Informix supports the following configurations:

- OnLine Workgroup Server, Version 7.2, and INFORMIX-Enterprise Command Center, Version 1.0
- OnLine Workgroup Server, Version 7.12, and INFORMIX-Enterprise Command Center, Version 2.0 or Version 1.0 ♦
- Dynamic Server, Workgroup and Developer Editions, Version 7.3, and Informix Enterprise Command Center, Version 3.0
- OnLine Workgroup Server, Version 7.22 through Version 7.24, and Informix Enterprise Command Center, Version 3.0
- OnLine Workgroup Server, Version 7.22, and INFORMIX-Enterprise Command Center, Version 2.0 ♦

Workgroup Edition 7.3 and OnLine Workgroup Server 7.22 through 7.24 and IECC 3.0

Scenario: Upgrade the database server and administration tools. This configuration supports all the new database server and administration tool functions. You can use IECC to display the version number and type of the database server.

OnLine Workgroup Server 7.12 and IECC 2.0

Scenario: Upgrade the administration tools but not the database server. With the 7.12 database server, you can use all the Version 2.0 administration tools except **Backup** and **Restore**.

OnLine Workgroup Server 7.22 and IECC 2.0

Scenario: Upgrade the database server but not the administration tools. This scenario could occur when your site has several client computers to upgrade over a period of time. Users would be able to use all the Version 2.0 administration tools but not GLS on the clients.

OnLine Workgroup Server 7.12 and IECC 1.0

Scenario: You decide not to upgrade the database server or administration tools.

Preparing for Migration

Use these instructions to migrate to Dynamic Server, Workgroup and Developer Editions, Version 7.3, from earlier versions on Windows NT or to OnLine Workgroup Server, Version 7.2, on UNIX. When you migrate a database server, Informix suggests that you follow these guidelines:

- Review the release notes for all versions of your database server for information about new features, installation, and fixes to problems. Modify applications as needed. Check the documentation notes for information about features that the manuals do not discuss.

The release notes are in one of the following directories:

- `$INFORMIXDIR/release/en_us/0333` ♦
- `%INFORMIXDIR%\release\en_us\0333` (as of Version 7.2)

Release notes appear in the Informix folder. To display this folder, choose **Start→Programs→Informix** from the Task Bar. ♦

- Check the machine notes for special actions you need to perform to configure and use your database server on your operating system. ♦
- On UNIX, retain both versions of the Informix product software on disk (if you have enough disk resources). On Windows NT, you cannot retain two versions of the Informix product on disk, prior to database server Version 7.3.
- Check the documentation notes for information about features that the manuals do not discuss.
- Retain the installation media from both versions of the Informix product software.
- Make a level-0 backup of the database server before and after migration.

UNIX

WIN NT

UNIX

WIN NT

Before you upgrade or revert your database server, complete the following steps, which the following sections describe in detail:

1. Check available space and system requirements.
2. Save a copy of the current configuration files.
3. Close all transactions and place the source database server in quiescent (administration) mode.
4. Verify the integrity of the data.
5. Back up the source database server.

Check Available Space and System Requirements

Before you install your database server and the administration tools, verify that your system meets the minimum space and hardware requirements.

Dynamic Server, Workgroup and Developer Editions, runs on Windows NT 3.51 and Windows NT 4.0 on an NTFS drive. IECC and the Relational Object Manager run on Windows NT 3.51, Windows NT 4.0, and on either a FAT or NTFS drive.

For information on the system requirements, refer to your *Installation Guide* and the **read_ows.txt** file in Answers OnLine, Version 1.7, and earlier.

Save a Copy of the Current Configuration Files

Save a copy of the current ONCONFIG files, located in the **etc** subdirectory of your installation directory, and the **sqlhosts** or registry information:

- \$ONCONFIG, **osahosts**, **sqlhosts**, **ttermcap**, **termcap** ♦
- %ONCONFIG% ♦

Save the following files, located in various subdirectories of your installation directory (this step is optional):

- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory

Close All Transactions and Place the Source Database Server in Quiescent Mode

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX. On Windows NT, you must be a member of the **Administration** group.
3. Execute the following command to take the database server to quiescent mode:

```
onmode -sy
```

4. Wait until your database server is in quiescent mode.

To verify the mode of your database server, execute the **onstat** - command. The first line of the **onstat** output contains the status of your database server.

Figure 14-5 shows that the database server is in quiescent mode.

```
INFORMIX-OnLine  Version  x.xx.xxx  --  Quiescent  --  Up  xx:xx:xx  --  xxxx Kbytes
```

OnLine Workgroup Server is in quiescent mode.

Figure 14-5
Example of onstat
Status Line

5. Execute the following command to force a new logical log:
6. Execute the following command to force a checkpoint:
7. Execute the following command to shut down the database server:

```
onmode -l
```

```
onmode -c
```

```
onmode -yuk
```

Tip: Monitor your log activity to verify that all commands were executed properly and to check for inconsistencies prior to migration.





To use IECC to place the system in administration mode

1. Select **Administration** from the **Server Mode** list box on the **General** page of Informix Enterprise Command Center to put the database server in quiescent (administration) mode.
2. Warn all users that you plan to shut down the database server and wait for them to exit. If users are logged on, the Administration wizard lets you notify users when their sessions are to be disconnected.

***Warning:** User data can be lost or damaged if you interrupt data transactions. If data transactions are interrupted, shut down and restart the database server in administration mode to initiate a fast recovery. A fast recovery rolls forward all committed transactions and rolls back all incomplete transactions since the last checkpoint and then leaves a new checkpoint record in the log with no open transactions pending.*

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data before you make a level-0 (complete) backup. If you find any problems with the data, fix them before you make the backup. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. First, you need to obtain a list of the databases on your database server.

To use oncheck on Windows NT

Open a **Command Line Utilities** window from the **Informix Administration Tools** program group. ♦

Figure 14-6 lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	tbcheck -cc <i>database_name</i>
Check data	tbcheck -cD <i>database_name</i>
Check indexes	tbcheck -cI <i>database_name</i>

Figure 14-6
*Commands for
Verifying the
Data Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

Back Up the Source Database Server

Make a complete backup of your source database server.

WIN NT

On Windows NT, double-click the **Backup and Restore** tool in the **Informix Enterprise Command Center** program group. You can also use the command line to back up with **ontape** and ON-Bar on Windows NT. ♦

UNIX

To launch Backup and Restore, enter **\$INFORMIXDIR/bin/bar** at the UNIX prompt. You can also use **ontape**, ON-Bar, or ON-Archive to back up on UNIX. ♦



Warning: The 7.12 and 7.22 and later versions of Backup and Restore are not compatible. The 7.12 version uses **ontape**, but the 7.22 and later versions use ON-Bar. Backups that you make under older versions of your database server are not compatible with the newer version. Do not try to restore these backups to the newer version.

For more information about how to use **ontape**, ON-Archive, or ON-Bar to back up your database server, refer to your *Backup and Restore Guide*.

Now you are ready to upgrade or revert your database server. For information on how to upgrade, see [“Upgrading to Dynamic Server, Workgroup and Developer Editions, Version 7.3.”](#) For information on how to revert, see [“Reverting the Database Server and IECC”](#) on page 14-38.

Upgrading to Dynamic Server, Workgroup and Developer Editions, Version 7.3

When you upgrade to Dynamic Server, Workgroup and Developer Editions, Version 7.3, you can install and test the database server with the same database server name, configuration files, environment variables, and shared-server computer that you used for the earlier version. After you install the target database server and verify that it works, you might want to modify the configuration files and environment variables to take advantage of new features.

Complete the following migration steps:

1. Save an output file of SQL statements for access paths.
2. Bring the source database server off-line.
3. Change UNIX kernel parameters.
4. Install the target database server and administration tools.
5. Customize the database server environment. (This step is optional.)
6. Prepare the database server for Enterprise Replication. (This step is optional.)
7. Install and configure SNMP.
8. Bring the target database server on-line.
9. Verify the integrity of the data.
10. Make an initial complete backup of the database server.
11. Run UPDATE STATISTICS.
12. Verify the access paths of your SQL statements.
13. Complete migration.

Save an Output File of SQL Statements for Access Paths

Save a file of output from any SET EXPLAIN statements. Later you can use this file to verify that access paths of your SQL statements do not change when you migrate to your target database server. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

WIN NT

For Windows NT, the SET EXPLAIN output filename is
`%INFORMIXDIR%\sqlexpln\username.out.` ♦

Bring the Source Database Server Off-Line

Shut down your database server to ensure that all common files are inactive.

The database server must be off-line because the older and newer versions share common files. You cannot install the database server if any of the common files are active.

WIN NT

On Windows NT, the installation program automatically shuts down the old database server and starts the new database server. ♦

UNIX

Change UNIX Kernel Parameters

You might need to change some of the kernel parameters for your UNIX operating system before you install the target database server. To reconfigure the operating system, follow the directions in the machine notes file included on your target database server distribution media and the kernel-configuration instructions for your operating system.

WIN NT

Install the Target Database Server and Administration Tools

You can upgrade the database server only, the administration tools only, or both. If a previous version of the database server or administration tools is on the computer, the **Upgrade** page appears when you install the new product. The Installation wizard replaces the files but does not reconfigure the database server.

To perform the upgrade

1. When the **Upgrade** page appears, click **Next** for the INFORMIX-OnLine Installation wizard. Click **Next** again to begin the installation procedure.
2. Enter the serial number and serial-number key.

The installation program automatically verifies and brings down OnLine Workgroup Server, copies the new files, and preserves the database and dbspace data. Then the installation program starts OnLine Workgroup Server, Version 7.22, with the same configuration and shared-server computer.

The Installation wizard updates the SQLHOSTS registry keys and OSAHOSTS information automatically.

Install and configure onsnmp

Workgroup Replication and the **onsnmp** utility require Windows NT SNMP. The installation and upgrade program checks the registry for the SNMP master agent. If the master agent was not installed, the program displays a warning message but does not configure the registry for Workgroup Replication or **onsnmp**. If you later choose to install the SNMP master agent, you must run the `%INFORMIXDIR%\bin\inssnmp.exe` command-line utility to install the SNMP subagents. You need not reinstall the database server.

Install the Target Database Server

On UNIX, you must be logged in as user **root**. On Windows NT, you must be a member of the **Administration** group to install your database server. Set the **INFORMIXDIR** environment variable to the directory where you plan to install your database server.

The installation script installs your database server into the **INFORMIXDIR** directory specified for user **root**. The installation script does not bring the database server on-line.

WIN NT



The setup program installs and brings up the database server on Windows NT. Follow the directions in your *Installation Guide* to install your database server. ♦

Warning: *If you install your database server in the same directory where the earlier version of the database server resides, the newer version overwrites the older files. If you want to preserve the files for the earlier version, you must install the newer database server in a different directory.*

Before you overwrite the older version, you must take the following precautions:

- If you do not have the original media for the older version, back up the **INFORMIXDIR** directory before you install your target database server.
- Copy the configuration file(s) in **INFORMIXDIR** in the **etc** directory to another location on the file system.

When you finish the installation and system reconfiguration, exit as user **root** and log on as user **informix**.

WIN NT

On Windows NT, you need to run the Installation wizard twice. First, upgrade the database server and then install the administration tools.

The Installation wizard replaces the files but does not reconfigure the database server. If a previous version of the database server is on the computer, the **Upgrade** page appears when you install the new product.

The installation program automatically verifies and brings down your source database server, copies the new files, and preserves the database configuration information. The installation program starts your target database server with the same configuration and shared-server computer. ♦

Install IECC

After you install the database server, install the administration tool, IECC.

WIN NT

After you install the database server, IECC, or both, the **Informix Administration Tools** program group is available in your Windows NT environment. For more information on installation, refer to your *Installation Guide*. ♦

Customize the Database Server Environment

After you install your database server, ensure that the following environment variables are set to the correct values:

- INFORMIXSERVER
- ONCONFIG
- PATH
- INFORMIXSQLHOSTS (if used)



Important: On UNIX, the client application looks for the *sqlhosts* file in the *etc* directory in the **INFORMIXDIR** directory. On Windows NT, **INFORMIXSQLHOSTS** points to the computer that contains the *sqlhosts* registry information. However, you can use the **INFORMIXSQLHOSTS** environment variable to change the location or name of the *sqlhosts* file.

WIN NT

On Windows NT, the installation program sets the configuration parameters and environment variables for you. However, you can customize the configuration parameters in the %ONCONFIG% file and environment variables for your database server. In Windows NT, use **setnet32** to customize the environment variables on the client computer. ♦

UNIX



For information about environment variables, refer to the *Informix Guide to SQL: Reference*.

Update the ONCONFIG Configuration Parameters

You can customize your **ONCONFIG** configuration file and environment variables to take advantage of the new features that Dynamic Server, Workgroup and Developer Editions, Version 7.3 introduced.

After you observe the performance of your database server, you might want to make further adjustments.

Important: Use the same values for your target database server for *ROOTOFFSET*, *ROOTSIZE*, and *ROOTPATH* that you used for your source database server. Also use the same values for size and number of physical logs, logical logs, and for mirroring (if available).

For information on how to configure, refer to your *Administrator's Guide*. For information about how to tune the configuration parameters, refer to your *Performance Guide*.

Update the Configuration Files

During the installation procedure, the install script checks the **etc** directory in the **INFORMIXDIR** directory for files named **config.arc**, **oper_deflt.arc**, **termcap**, **logevent.sh**, **sessalrm**, and **permalrm**. If these files do not exist, the install script provides them. If the files do exist, the install script does not overwrite the files. Instead, the install script provides additional files named **Config.arc**, **Oper_deflt.arc**, **Termcap**, **Logevent.sh**, **Sessalrm**, and **Permalrm** (note the initial uppercase letters).

Compare your current versions of the files with the new versions and determine whether new or changed configuration parameters or qualifiers exist.

Prepare the Target Database Server to Use Enterprise Replication

Perform this step only if you plan to use Enterprise Replication with your database server (Version 7.22, or later).

Before you can activate Enterprise Replication, you need to update the **sqlhosts** file or registry key for each database server that participates in Enterprise Replication.

To activate Enterprise Replication for the first time

1. Bring your database server off-line.
2. Define a **group name** for Enterprise Replication.
3. Assign a **group identifier** to the group and include this group ID in the **sqlhosts** option key or column
4. Bring your database server on-line.
5. For upgrades to Versions 7.22 to 7.24, verify that the SNMP service is installed on the host and that the SNMP subagents are registered.

In Dynamic Server 7.30, you need SNMP only if you plan to use the Enterprise Replication Monitoring Program. (Version 7.31 does not include the Enterprise Replication Monitoring Program.)

6. Use the Replication Manager to define each database server for replication.

This step starts Enterprise Replication.

When you upgrade to Dynamic Server 7.3x, the conversion upgrades the **syscdr** database and the **sysmaster** and **sysutils** databases and system tables to Version 7.3x.

When you upgrade to a Version 7.3x database server from a Version 7.2x database server, the Enterprise Replication state is the same in Version 7.3x as it was in Version 7.2x.

For information on how to configure Enterprise Replication, refer to the *Guide to Informix Enterprise Replication*. For information on how to use SNMP, refer to the *Informix SNMP Subagent Guide*.

WIN NT

Install and Configure SNMP

In Dynamic Server 7.30, if you plan to use the Enterprise Replication Monitoring Program, you must install SNMP. Enterprise Replication in Dynamic Server 7.30 does not require SNMP. The SNMP management tools use the **OnSnmpSubagent** to respond to queries.

For Informix database servers Version 7.22 through Version 7.24, if you use Workgroup Replication on Windows NT, the **onsnmp** utility requires Windows NT SNMP. The installation and upgrade program checks the registry for the SNMP master agent. If the master agent was not installed, the program displays a warning message but does not configure the registry for Workgroup Replication or **onsnmp**. If you later choose to install the SNMP master agent, you must run the %INFORMIXDIR%\bin\inssnmp.exe command-line utility to install the SNMP subagents. You do not need to reinstall the database server.

Bring the Target Database Server On-Line

When you bring your target database server on-line for the first time, bring it first to quiescent mode and then to on-line mode.

On UNIX, execute the following command to bring your database server from off-line to quiescent mode:

```
oninit
```

Once the database server is in quiescent mode, check the message log for status messages.

Important: *If you note problems in the message file, solve the problems before you continue to the next step.*

Execute the following command to change your database server mode from quiescent mode to on-line mode:

```
onmode -m
```

The **sysmaster** and **sysutils** databases are created once your database server is brought on-line.

Warning: *The logical logs continue to fill with the transactions that result from the creation of the **sysmaster** database. If you run out of log space before the creation of the **sysmaster** database is complete, the database server halts with a “Logs Full” error. Thus, you must back up the logical logs.*

Use ON-Bar, ON-Archive, or **ontape** to back up logical logs. ♦

On Windows NT, the setup program brings your database server on-line automatically. ♦

UNIX



WIN NT

If you customized the database server environment, bring down and restart the database server with IECC. When you restart your database server, the changes to the configuration parameters and environment variables take effect.

To start your database server with IECC

1. In the IECC **Console** program group, double-click the Informix Enterprise Command Center icon.
2. In IECC, select the database server in the tree view.
3. Choose **Server→On-line**.

You can start the database server with or without the IECC interface. For more information, refer to the *Informix Enterprise Command Center User Guide*. ♦

To start your database server with IECC

1. In the **INFORMIX-OnLine Workgroup Server** program group, double-click the **Command Center** icon. In the **Command Center**, choose a database server from the **Servers** menu.
2. On the **General** page, select **On-Line** in the **Server Mode** list box.

Verify the Integrity of the Data

Before you allow users to access the database, use the **oncheck** utility to verify that no data was corrupted in the migration process. You can verify the integrity of the reserve pages, extents, system catalog tables, data, and indexes. For more information, see [“Verify the Integrity of the Data” on page 14-26](#).

Make an Initial Complete Backup of the Target Database Server

Use IECC to make a complete, whole-system backup of your database server. For more information, see the *Informix Enterprise Command Center User Guide*.

Use your database server backup utility (ON-Bar, ON-Archive, or **ontape**) to make a level-0 backup. Do not overwrite the tapes you used earlier when you made the final backup of your database server. For more information about how to make a backup, refer to your *Archive and Backup Guide* or your *Backup and Restore Guide*.



Important: Do not restore the backed-up logical logs to the newer version of your database server from the earlier version of your database server.

Run UPDATE STATISTICS

Run the UPDATE STATISTICS statement according to the recommended procedure in the *Informix Guide to SQL: Syntax*. UPDATE STATISTICS updates the information that your database server uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Verify the Access Paths of Your SQL Statements

Use the SET EXPLAIN statement to verify that the access paths of your SQL statements did not change when you migrated to your target database server. If you have SET EXPLAIN output from the source database server, run SET EXPLAIN for your target database server. Compare the SET EXPLAIN output from both the source and target database servers. SET EXPLAIN writes the access path that the optimizer chooses for each query to the SET EXPLAIN output file. The optimizer chooses the fastest path of execution for table joins.

UNIX

The UNIX SET EXPLAIN output filename is **sqexplain.out**. ♦

WIN NT

For Windows NT, the SET EXPLAIN output filename is %INFORMIXDIR%**sqlexpln\<username>.out**. ♦

If the SET EXPLAIN output file shows that a different access path was used, complete the following steps:

1. Check the **OPTCOMPIND** environment variable or configuration parameter.
2. Check the **DBSPACETEMP** environment variable or configuration parameter to ensure that adequate temporary dbspaces are defined. You might need to define more temporary dbspaces.
3. Analyze the query access paths and modify the schema to improve the performance, if necessary.

Complete Migration

The first time your database server is brought on-line, the **sysmaster** database is built. Check the message log to ensure that the **sysmaster** database build is complete before you allow users to access the database server. After you complete a level-0 backup and you ensure that client users can access data on your database server, the migration process is complete.

After you successfully upgrade to Dynamic Server, Workgroup and Developer Editions, Version 7.3, you might want to seek ways to obtain maximum performance. If you created sample queries for comparison, you can use them to characterize the performance differences between your earlier version and your new version. The results of these comparisons might suggest adjustments to configuration parameters or to the layout of databases, tables, and chunks. For details on topics related to performance, refer to your *Performance Guide*.

WIN NT

Changing Database Server Definitions on Windows NT

The installation program automatically updates the registry information and the database server definitions on the shared-server (SQLHOSTS) computer. The *database server definitions* consist of the database server name, TCP/IP host names, network types, and the service names.

Use IECC if you want to change the **informix** user password, specify a different computer as the shared-server computer, or edit database server definitions. For example, you might specify a different shared-server computer if you have migrated the database server to a new computer or connected a single client to several database servers.

For information on how to use the **Client Control Panel** folder in IECC to configure database servers, see the *Informix Enterprise Command Center User Guide*. For information on connectivity, see your *Administrator's Guide*.

Migrating to a GLS Locale

If you want to migrate your database server to a non-English GLS locale, set the `DB_LOCALE` and `CLIENT_LOCALE` environment variables before you open the database in Version 7.3. If your previous database server version used Native Language Support (NLS), replace the NLS environment variables, such as `COLLCHAR`, with GLS environment variables. For information on how to work with locales and how to set GLS environment variables, see the *Informix Guide to GLS Functionality* and Section VI “[Locale Changes](#).”

Reverting the Database Server and IECC

This section describes the steps for reverting to an earlier version from Dynamic Server, Workgroup and Developer Editions, Version 7.3, and IECC 3.0.

Follow the preparatory steps, described in “[Preparing for Migration](#)” on [page 14-23](#) and then complete the following steps:

1. Remove unsupported SQL features.
2. Uninstall Enterprise Replication or Workgroup Replication, if it is installed.
3. Run the reversion utility (**onmode -b**).
4. Uninstall Dynamic Server, Workgroup and Developer Editions, and IECC.
5. Remove GLS features (only if GLS was used and you are reverting to Version 7.12).
6. Modify configuration parameters.
7. Reset environment variables.
8. Reinstall the old version of the database server and IECC.
9. Bring the target database server on-line.
10. Verify the integrity of the data.
11. Make an initial complete backup of the target database server.
12. Complete reversion.

The following sections describe these steps in detail.

Remove Unsupported SQL Features

Before you revert, you must remove SQL features that the earlier version of your database server does not support. See the “New Features of This Product” section in the appropriate version of the *Informix Guide to SQL: Syntax*.

Uninstall Enterprise Replication or Workgroup Replication, Version 7.3

Skip this section if Enterprise Replication or Workgroup Replication is not installed on your system.

To revert to an earlier version if Enterprise Replication or Workgroup Replication is active

1. Stop Enterprise Replication or Workgroup Replication.
2. For altered tables with CRCOLS, issue the command:


```
alter table drop CRCOLS
```

◆
3. Execute the **onmode -b** command to revert to the earlier version of your database server.

Warning: If you try to revert to a previous version of the database server while Enterprise Replication is active, the reversion will fail.

To revert to an earlier version if Enterprise Replication or Workgroup Replication is inactive

1. In this situation, Enterprise Replication or Workgroup Replication was previously active on this database server.
2. For altered tables with CRCOLS, issue the command:


```
alter table drop CRCOLS
```

◆
3. Execute the **onmode -b** command to revert to the earlier version of the database server. The **syscdr** database is dropped during reversion.

For more information, see the *Guide to Informix Enterprise Replication*.

UNIX



UNIX

Run the Reversion Utility

The database server must be running when you execute the reversion utility. Execute the reversion utility to revert the database, where *version_number* is the earlier database server version (see [Figure 14-7](#)), as follows:

```
onmode -b version_number
```

After the reversion is complete, your database server is off-line. For more information about the **onmode -b** command, refer to [Chapter 24, “Using the onmode Utility.”](#)

Revert To	Revert From	Command
Version 7.22	Version 7.3	onmode -b 7.22
Version 7.2	Version 7.22	onmode -b 7.2
Version 7.12	Version 7.2x	onmode -b 7.1

Figure 14-7
Reverting to an Earlier
Database Server

Uninstall the Database Server and IECC

Uninstall both the database server and IECC.

To uninstall the product

- 1. Double-click the **Uninstall** icon in the **IECC Console** program group.
- 2. In the Uninstall dialog box, check **Remove OnLine Server**.

Warning: Do not check **Remove all OnLine databases, supporting files and all database information**. If you check this option, your configuration, dbspaces, and database information will be lost, making reversion impossible.

- 3. To uninstall the administration tools, check **Remove Administration Tools**.
- 4. Click **OK** to uninstall the database server.

For more information, see your *Installation Guide*.

WIN NT



GLS***Remove GLS Features (Only If Reverting to 7.12)***

Skip this step if your database server uses the default English locale (**en_us.8859-1**). To revert the database server to Native Language Support (NLS) or Asian Language Support (ALS) from GLS, set the appropriate NLS or ALS locales and environment variables. For information on how to work with locales, see Section VI, “[Locale Changes](#),” and the *Informix Guide to GLS Functionality*.

Modify Configuration Parameters

Dynamic Server, Workgroup and Developer Editions, Version 7.3, uses configuration parameters that did not appear in earlier versions of the database server that you should remove or modify prior to reversion.

For more information on these configuration parameters, see “[New Configuration Parameters in Dynamic Server 7.3](#)” on page 11-7.

Configuration Changes for Version 7.12

You must add the ONLANGMAP configuration parameter to your ONCONFIG file before you start OnLine Workgroup Server 7.12. ONLANGMAP takes the value **en_US-English**. For information on ONLANGMAP, see the OnLine Dynamic Server for Windows NT, Version 7.12.TC2, documentation notes. ♦

If you are reverting to Version 7.12, remove the HETERO_COMMIT configuration parameter from your ONCONFIG file.

Configuration Changes for Version 7.22

You might need to revise the value of the ALARMPROGRAM configuration parameter in your configuration file.

WIN NT

If you used Enterprise Replication and are reverting to Version 7.12 or Version 7.2, remove the following configuration parameters from your ONCONFIG file:

- CDR_LOGBUFFERS
- CDR_EVALTHREADS
- CDR_DSLOCKWAIT
- CDR_QUEUEMEM
- CDR_NIFUSEHELP
- CDR_NIFMEMS
- SCDR_NIFQUEUES

For more information on these configuration parameters, see the *Guide to Informix Enterprise Replication*.

Reset Environment Variables

Reset the environment variables to values that are appropriate for your version of your database server. For information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Environment Variable Changes for Version 7.12

OnLine Workgroup Server, Version 7.12, supports NLS but does not support GLS. When you revert to Version 7.12, delete the following environment variables:

- CC8BITLEVEL
- CLIENT_LOCALE
- DBCENTURY
- DBFLTMASK
- DBONPLOAD
- DB_LOCALE
- ESQLMF
- GLS8BITFSYS
- GL_DATES

NLS

UNIX

- GL_DATETIME
- ONPLOAD
- PLCONFIG
- SERVER_LOCALE
- THREADLIB

Add the following environment variables:

- DBNLS
- COLLCHAR
- LANG

Reinstall the Old Version of the Database Server

On UNIX, you do not need to reinstall the administration tools because they were not upgraded for Version 7.2.

To perform the installation

1. Supply customer information.
2. On the **OnLine Workgroup Server Installation Options** screen, choose option 1, **Install**.
3. On the **Run Setup Again** screen, choose option 1, **Install All Files, But Leave Configuration Alone**. This option copies the new database server files without changing the configuration and **sqlhosts** file information.
4. The installation program automatically verifies and brings down the database server, copies the new files, and saves the database and dbspace data. It starts the earlier version of your database server with the old configuration.



Warning: Do not select option 2, **Install OnLine Workgroup Server and Reconfigure**. If you select option 2, your configuration and database information will be lost. ♦

WIN NT

On Windows NT, reinstall the earlier version of the database server in the same directory as the files for the previous version.



UNIX

WIN NT

To perform the installation

1. On the **Run Installation Again** page, select the **Copy all files, but leave configuration alone** installation option.
2. Supply your serial number and serial number key, as shown on the serial-number key card.
3. Select one or both of the components you want to install: **OnLine Database Server** or **Administration Tools**. Click **Next**.
4. The installation program automatically copies the older database server files and saves the configuration and the database data.

Warning: Do not select **Copy all files and reconfigure the product**. If you select this option, your configuration and database information will be lost.

For more information on installation, refer to the *INFORMIX-Enterprise Command Center User Guide*.

Warning: If you are reverting to database server Version 7.12, you must add the *ONLANGMAP* parameter to the *ONCONFIG* file before you start the reverted database server. *ONLANGMAP* takes the value **en_US-English** and ensures backward compatibility with non-NLS locales. If the database server fails on install, just restart it. ♦

Bring the Target Database Server On-Line

On UNIX, execute the following command to bring your database server to quiescent mode:

```
oninit
```

♦

On Windows NT, the installation program brings the database server on-line automatically. ♦

The database server initializes the shared memory and builds the **sysmaster** database. After the **sysmaster** database is built, the reversion process is complete.

Verify the Integrity of the Data

To verify the integrity of your data, follow the steps as described in [“Verify the Integrity of the Data” on page 14-26](#).

Make an Initial Complete Backup of the Target Database Server

Informix recommends that you make a complete, level-0 backup of your database server after you complete the reversion. You can use the ON-Bar, ON-Archive, or **ontape** utility to make the backup. For information about how to make a backup, see [“Back Up the Source Database Server” on page 14-27](#) or refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.



Important: Do not overwrite the tapes that you used to back up your database server.

Complete Reversion

After the **sysmaster** and **sysutils** databases are built, the reversion process is complete. Ensure that client users can access data on the earlier version of your database server.

Moving Data Between SE and Dynamic Server 7.3 or Its Editions

This section describes the following procedures:

- Moving data to Dynamic Server 7.3 or its editions from SE
- Moving data to SE from Dynamic Server 7.3 or its editions

Moving Data to Dynamic Server 7.3 or Its Editions from SE

The following sections describe the steps for moving data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE. To move the data, you need to choose a migration method, eliminate SE-specific features, migrate the data from SE, and modify client applications.

Using the dbexport and dbimport Utilities

If you intend to move an entire database to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE, the **dbexport** and **dbimport** combination is the easiest migration method.

To use **dbexport** and **dbimport** to move data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE

1. Use the **dbexport** utility to export the data from SE.

You can move the data to another directory or directly to tape.

Warning: Do not use the **-ss** option with **dbexport** when you move data between database servers. The **-ss** option generates SE-specific syntax that Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, does not recognize.

2. Edit the database schema file (the **.sql** file that **dbexport** creates).

You might want to add information that Dynamic Server 7.3 (or Dynamic Server, Workgroup and Developer Editions) databases and tables can use, such as the following data:

- Initial- and next-extent values for a table
 - Lock mode for a table
 - Blobspace where TEXT or BYTE data types should reside
 - Dbospace where the tables should reside
 - VARCHAR or NVARCHAR column specifications
 - Fragmentation schemes
3. You can also make the following changes to the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes.
 - Specify the dbospace location of the database.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary. (You can write a script to automate the task of modifying the SQL statements.)

For details about supported SQL statements, see the *Informix Guide to SQL: Syntax*.





4. If necessary, install the new database server. For installation procedures, see your *Installation Guide*.

Important: For Dynamic Server 7.3 or OnLine Dynamic Server 7.1 or 7.2, you must access rows that were fragmented using column values (primary key values) in a row rather than ROWID.

5. Change the **INFORMIXSERVER** environment variable to specify your target database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Start the new database server.
8. Use the **dbimport** utility to move the database data into the new database server.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 20, “Using the dbexport and dbimport Utilities.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows into a text file.

Using UNLOAD, LOAD, and dbschema

If you want to move selected tables or columns instead of an entire SE database to your target database server, use the SQL statements UNLOAD and LOAD in the DB-Access utility with the **dbschema** database utility.

To use UNLOAD, LOAD, and dbschema to move data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE

1. Make sure that you have sufficient disk space to store the unloaded data. (You cannot unload data to tape with UNLOAD.)
2. Invoke the DB-Access utility.
3. Use the UNLOAD statement to move the selected data into a text file. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.

5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the SE database.
 - b. Edit the schema file so that it describes the new tables.
 - c. Make any of the following changes to the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes.
 - Specify the dbspace location for the table.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary.

For details about supported SQL statements, see the *Informix Guide to SQL: Syntax*.

If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.

6. If necessary, install your target database server.
7. Change the **INFORMIXSERVER** environment variable to specify your target database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Start the new database server.
10. Invoke the DB-Access utility.
11. Select the target database.

If you are creating a new database, execute the **CREATE DATABASE** statement or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table or tables, choose and run the schema file that you prepared in step 5 or enter **CREATE TABLE** statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. Use **LOAD** statements to load the data into the desired tables.

Using UNLOAD, dbload, and dbschema

If you need to manipulate the data before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbload, and dbschema to move data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE

1. Follow steps 1 through 13 of “Using UNLOAD, LOAD, and **dbschema**” on page 14-47.
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to Section VIII, “Data Migration Utilities.” For information on how to use DB-Access, refer to the *DB-Access User Manual*.

Moving C-ISAM Data to Dynamic Server or Its Editions

After you convert the C-ISAM data files to SE format, use **dbexport** and **dbimport** to move the data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, from SE. To convert the data files to SE format, follow the procedure under “Converting C-ISAM Files to SE” on page 13-6. For more information on how to work with C-ISAM files, see the *C-ISAM Programmer’s Manual*.



Tip: For details on how to convert C-ISAM data files to SE, contact your local systems engineer for assistance.

Adapting Programs for Dynamic Server 7.3 or Its Editions

After you successfully move the SE data to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, verify that your application developers know the differences between both database servers. For more information about the server-specific limitations of SQL statements, refer to the *Informix Guide to SQL: Syntax* and the *Informix Guide to SQL: Reference*. For information about environment variables, see the *Informix Guide to SQL: Reference*.

SE

For more information about how to configure your database server, refer to your *Administrator's Guide*.

The following statements contain syntax and keywords that only SE recognizes:

- CHECK TABLE
- CREATE AUDIT
- DROP AUDIT
- RECOVER TABLE
- REPAIR TABLE
- ROLLFORWARD DATABASE
- START DATABASE ♦

Moving Data to SE from Dynamic Server or Its Editions

When you move data to SE from Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, you must choose a migration method, eliminate features that SE does not support, migrate the data, and modify applications.

Eliminating Features That SE Does Not Support

Before you export your source database server data to SE, you must eliminate or upgrade the following unsupported data types to a data type that SE supports:

- VARCHAR
- BYTE
- TEXT

Using the dbexport and dbimport Utilities

If you intend to move an entire database to SE, the **dbexport** and **dbimport** combination is the easiest migration method.



To use `dbexport` and `dbimport` to move data to SE from your source database server

1. Use **`dbexport`** to export the data from your source database server.

You can move the data to a directory or directly to tape.

Warning: Do not use the **`-ss`** option with **`dbexport`** when you move data between database servers. The **`-ss`** option generates syntax that is specific to Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions, that SE does not recognize.

2. Remove the following information from the CREATE TABLE statements in the schema file (the **`.sql`** file that **`dbexport`** creates):
 - Initial- and next-extent sizes
 - Lock modes
 - Dbspace names
 - Blobspace names
 - Logging modes
 - Table-fragmentation expressions
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges on specific tables and indexes
 - Change the logging status of the database
 - Change the ANSI-compliance status of the database
4. If necessary, install SE. For installation instructions, refer to your *Installation Guide*.
5. Change the **`INFORMIXSERVER`** environment variable to specify your new database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **`DBDATE`** and **`DBMONEY`** formats.
7. Move to the directory where **`dbimport`** will store the SE database.
8. Use **`dbimport`** to move the data into an SE database.

For detailed descriptions of the **`dbexport`** and **`dbimport`** utilities, refer to [Chapter 20, “Using the `dbexport` and `dbimport` Utilities.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows into a text file.

Using UNLOAD, LOAD, and dbschema

If you want to move selected tables or columns to SE instead of moving an entire database, use the UNLOAD and LOAD statements in the DB-Access utility with the **dbschema** utility.

To use UNLOAD, LOAD, and dbschema to move data to SE

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. Use UNLOAD statements to move the selected data into text files. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the source database server.
 - b. Edit the schema file so that it describes the new tables.
6. If necessary, install SE.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.
10. Select the target database.

If you are creating a new database, execute the CREATE DATABASE statement or choose **Database→Create** from the DB-Access menu.

11. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5 or enter CREATE TABLE statements to create the new tables.
12. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
13. Use LOAD statements to load the data into the desired tables.

Using UNLOAD, dbload, and dbschema

If you need to manipulate the data before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbload, and dbschema to move data to SE from Dynamic Server 7.3 or Dynamic Server, Workgroup and Developer Editions

1. Follow steps 1 through 12 of “Using UNLOAD, LOAD, and [dbschema](#)” on page 14-52.
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to Section VIII, “[Data Migration Utilities](#).” For information on how to use DB-Access, refer to the *DB-Access User Manual*.

Adapting Your Programs for SE

After you migrate data to SE, verify that your application developers know the differences between both database servers.

SE

SE does not use the ONCONFIG configuration file. SE supports a subset of the environment variables that Dynamic Server and Dynamic Server, Workgroup and Developer Editions, support. For more information about how to use SE, refer to the *INFORMIX-SE Administrator's Guide*. ♦

For more information about the differences between database servers and their interpretation of SQL, refer to the *Informix Guide to SQL: Reference*.

IDS 7.3

Dynamic Server 7.3 supports the following SQL statements, which SE does not support:

- ALTER FRAGMENT
- GRANT FRAGMENT
- REVOKE FRAGMENT
- SET DATASKIP
- SET PDQPRIORITY ♦

IDS 7.3

W/D

Dynamic Server 7.3 and Dynamic Server, Workgroup and Developer Editions, support the following SQL statements, which SE does not support:

- CREATE ROLE
- DROP ROLE
- RENAME DATABASE
- SET ISOLATION
- SET LOG
- SET ROLE
- SET SESSION AUTHORIZATION ♦

If you change the name or pathname of the database server, update the **DBPATH** environment variable with the location of the new database. For information about **DBPATH**, refer to the *Informix Guide to SQL: Reference*.

Migration Between Operating Systems

- Chapter 15** **Moving a Database Server to a Different Operating System**
- Chapter 16** **Migrating Between Dynamic Server 7.3 and Its Editions in Different Environments**



Moving a Database Server to a Different Operating System

In This Chapter	15-3
Choosing a Migration Method	15-3
Adjusting Database Tables for File-System Variations	15-5
Moving Data to a Database Server on a Different Operating System	15-5
Using the Migration Utilities	15-6
Adapting Your Programs for UNIX or Windows NT	15-7
Completing Migration	15-7



In This Chapter

This chapter describes the steps for migrating between Informix database servers in different operating systems. The description covers the following topics:

- Choosing a migration method
- Adjusting database tables for file-system variations
- Migrating a database server to a different operating system

[Figure 1-6 on page 1-11](#) shows the paths for migrating to a different operating system.

Choosing a Migration Method

UNIX and Windows NT store data in different page sizes. When your migration involves different operating systems, you must export data and its schema information from one database server and import the exported data into the other database server.

The method that you choose for exporting and importing data depends on how much data you plan to move. All these methods deliver similar performance and enable you to modify the schema of the database. You can use the following migration methods:

- **dbexport** and **dbimport**

To move an entire database, use the **dbexport** and **dbimport** utilities.

- **UNLOAD** and **LOAD**

To move selected columns or tables, use the **UNLOAD** statement. Use **LOAD** when you do not want to change the data format.

- **UNLOAD**, **dbload**, and **dbschema**

To move selected columns or tables, use the **UNLOAD** statement. Use **dbload** to change the data format.

- **onunload** and **onload**

To unload data in page-sized chunks, use the **onunload** utility. Use the **onload** utility to move data to an identical database server on a computer of the same type.

- The High-Performance Loader (HPL)

To move selected columns or tables or an entire database, use the HPL.

- External Tables

To load and unload data in Dynamic Server with AD and XP Options or OnLine XPS, use external tables. For more information on this topic, see [“External Tables” on page 2-19](#) and [“Loading and Unloading Data” on page 10-19](#). ♦

Important: Do not use **onunload** and/or **onload** with Dynamic Server 2000, Universal Server, Extended Parallel Server, or Dynamic Server with AD and XP Options data. Instead, use the HPL for Dynamic Server 2000 or Universal Server, **onxfer** for Extended Parallel Server, or external tables for Dynamic Server with AD and XP Options.

AD/XP

XPS 8.11



Adjusting Database Tables for File-System Variations

File-system limitations vary between NFS and non-NFS file systems. You might need to break up large tables when you migrate to a new operating system.

For example, if you have a 3-gigabyte table, but your operating system only allows 2-gigabyte files, break up your table into separate files before you migrate. For more information, see your *Administrator's Guide*.

Moving Data to a Database Server on a Different Operating System

This section describes the steps for moving data between Informix database servers in UNIX and Windows NT environments.

To move data to a database server on a different operating system

1. Save a copy of the current configuration files. For detailed steps, see [“Save a Copy of the Current Configuration Files” on page 14-9](#).
2. Use ON-Bar, ON-Archive, or **ontape** to make a final level-0 backup. For more information, refer to your *Backup and Restore Guide* or your *Archive and Backup Guide*.

Dynamic Server with AD and XP Options, Extended Parallel Server, and OnLine XPS do not support ON-Archive or **ontape**. ♦

3. Choose one of the following sets of migration utilities to unload the databases:
 - **dbexport** and **dbimport**
 - UNLOAD, **dbschema**, and LOAD
 - UNLOAD, **dbschema**, and **dbload**
 - External tables with **dbschema** (for Dynamic Server with AD and XP Options or OnLine XPS) ♦
4. Bring the source database server off-line.

AD/XP

XPS 8.3

XPS 8.11

AD/XP

XPS 8.11

5. Install and configure the target database server. If you are migrating to Windows NT, also install the administration tools.
6. Bring the target database server on-line.
7. Use **dbimport**, **LOAD**, or **dbload**, or external tables to load the databases into the target database server, depending on which utility you used to export the databases.
8. Make an initial level-0 backup of the target database server.
9. Run **UPDATE STATISTICS** to update the information that the target database server uses to plan efficient queries.

Using the Migration Utilities

Choose one of the following migration utilities:

- If you intend to move an entire database on Dynamic Server 2000 or Dynamic Server 7.3 between different environments, the **dbexport** and **dbimport** combination is the easiest migration method. Follow the steps in [“Using the dbexport and dbimport Utilities” on page 14-46](#). ♦
- If you want to move selected tables or columns, instead of an entire database, use the **UNLOAD** and **LOAD** statements with the **dbschema** utility. Follow the steps in [“Using UNLOAD, dbschema, and LOAD” on page 14-6](#).
- If you need to manipulate the data in the specified **UNLOAD** file before you load it into a new table, use a combination of the **UNLOAD** statement and the **dbschema** and **dbload** utilities. Follow the steps in [“Using UNLOAD, dbschema, and dbload” on page 14-8](#).
- For Dynamic Server with AD and XP Options or OnLine XPS, use external tables, as the *Administrator’s Reference* describes. ♦

IDS 2000

IDS 7.3

AD/XP

XPS 8.11

Adapting Your Programs for UNIX or Windows NT

Certain database server configuration parameters and environment variables are environment dependent, as follows:

- Dynamic Server 2000 supports Enterprise Replication.
- Dynamic Server 7.3 supports Enterprise Replication and uses Version 3.0 of Informix Enterprise Command Center (IECC).
- Dynamic Server, Workgroup and Developer Editions, for UNIX is at Version 7.2. It supports GLS and uses Version 1.0 of IECC.
- Dynamic Server, Workgroup and Developer Editions, for Windows NT is at Version 7.3. It supports GLS, ON-Bar, Enterprise Replication, and the Gateway products and uses Version 3.0 of IECC.

For details, see [Appendix A](#), your *Administrator's Guide*, or the *Administrator's Reference*.

Completing Migration

The first time the target database server is brought on-line, the **sysmaster** and **sysutils** databases are built. Check the message log to ensure that the **sysmaster** and **sysutils** databases have been created successfully before you allow users to access the database server. After you ensure that client users can access data on the database server, the migration process is complete.

Then you might want to seek ways to obtain maximum performance. For details on topics related to performance, refer to your *Performance Guide*.

Migrating Between Dynamic Server 7.3 and Its Editions in Different Environments

In This Chapter	16-3
Choosing a Migration Method	16-4
Migrating Data to Dynamic Server, W/D from 7.3 on a Different Operating System	16-5
Save Copies of the Current Configuration Files	16-6
Verify the Integrity of the Data	16-6
Back Up Dynamic Server 7.3	16-7
Export Data from Dynamic Server 7.3	16-7
Using the dbexport and dbimport Utilities	16-8
Using the UNLOAD Statement	16-9
Shut Down Dynamic Server 7.3	16-11
Install and Configure Dynamic Server, Workgroup and Developer Editions	16-12
Verify Port Numbers and the Services File	16-12
Customize the Database Server Environment	16-12
Bring Dynamic Server, Workgroup and Developer Editions, On-Line.	16-13
Import Data into Dynamic Server, Workgroup and Developer Editions	16-13
Verify the Integrity of the Data	16-14
Make an Initial Backup of Dynamic Server, Workgroup and Developer Editions	16-14
Run UPDATE STATISTICS	16-14
Complete Migration	16-15
Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions	16-15

Migrating Data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on a Different Operating System.	16-15
Save Copies of the Current Configuration Files	16-16
Verify the Integrity of the Data	16-17
Back Up Dynamic Server, Workgroup and Developer Editions	16-17
Export Data from Dynamic Server, Workgroup and Developer Editions	16-17
Using the dbexport and dbimport Utilities	16-18
Using the UNLOAD Statement	16-19
Shut Down Dynamic Server, Workgroup and Developer Editions	16-20
Install and Configure Dynamic Server 7.3	16-21
Verify Port Numbers and the Services File	16-21
Customize the Database Server Environment	16-21
Bring Dynamic Server 7.3 On-Line	16-22
Import the Data into the Target Database Server	16-22
Verify the Integrity of the Data	16-22
Back Up the Target Database Server	16-23
Run UPDATE STATISTICS	16-23
Complete Migration	16-23
Adapt Your Programs for Dynamic Server 7.3	16-24

In This Chapter

This chapter describes the steps for migrating between Dynamic Server, Version 7.3, and its editions in different operating systems. [Figure 1-6 on page 1-11](#) shows the paths for migrating to a different operating system. This chapter covers the following topics:

- Choosing a migration method
- Migrating data to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3 on a different operating system
- Migrating data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on a different operating system

Choosing a Migration Method

UNIX and Windows NT store data in different page sizes. When your migration involves different operating systems, you must export data and its schema information from one database server and import the exported data into the other database server.

The method that you choose for exporting and importing data depends on how much data you plan to move. All these methods deliver similar performance and enable you to modify the schema of the database. You can use the following migration methods:

- **dbexport** and **dbimport**
To move an entire database, use the **dbexport** and **dbimport** utilities.
- **UNLOAD** and **LOAD**
To move selected columns or tables, use the **UNLOAD** statement. Use **LOAD** when you do not want to change the data format.
- **UNLOAD** and **dbload**
To move selected columns or tables, use the **UNLOAD** statement. Use **dbload** to change the data format.
- **onunload** and **onload**
To unload data in page-sized chunks, use the **onunload** utility. Use the **onload** utility to move data to an identical database server on a computer of the same type.
- The High-Performance Loader (HPL)
To move selected columns or tables or an entire database, use the HPL.

Migrating Data to Dynamic Server, W/D, from 7.3 on a Different Operating System

The following sections describe the steps for moving data to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3 on a different operating system.

When you move data between UNIX and Windows NT, you must choose a migration utility, eliminate database server- and environment-specific features, migrate the data, and modify applications. Complete the following migration steps, which later subsections describe in detail:

1. Save copies of the current configuration files.
2. Verify the integrity of the data.
3. Make a final backup of Dynamic Server 7.3.
4. Export data from Dynamic Server 7.3.
5. Shut down Dynamic Server 7.3.
6. Install and configure Dynamic Server, Workgroup and Developer Editions.
7. Verify port numbers and the **services** file.
8. Customize the database server environment (optional).
9. Bring Dynamic Server, Workgroup and Developer Editions, on-line.
10. Import data into Dynamic Server, Workgroup and Developer Editions.
11. Verify the integrity of the data.
12. Make an initial backup of Dynamic Server, Workgroup and Developer Editions.
13. Run UPDATE STATISTICS.
14. Complete migration.
15. Adapt your programs for Dynamic Server, Workgroup and Developer Editions.

Save Copies of the Current Configuration Files

Save copies of the current configuration files that you modified. These should include the following files:

- Current **ONCONFIG** file, located in the **etc** subdirectory of your installation directory
- **sqlhosts** information
- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory
- ON-Archive configuration files, located in the **etc** subdirectory

If you use ON-Bar to back up your source database server and the logical logs, you also need to save a copy of the following file:

UNIX: **\$INFORMIXDIR/etc/ixbar.<servername>**
Windows NT: **\$INFORMIXDIR\etc\ixbar.<servername>**

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of the data. [Figure 16-1](#) lists the commands for verifying data integrity.

Action	oncheck Command
Check reserve pages	oncheck -cr
Check extents	oncheck -ce
Check system catalog tables	oncheck -cc <i>database_name</i>
Check data	oncheck -cD <i>database_name</i>
Check indexes	oncheck -cI <i>database_name</i>

Figure 16-1
*Commands for
Verifying Data
Integrity*

For information about **oncheck**, refer to your *Administrator's Guide*.

UNIX

WIN NT

Back Up Dynamic Server 7.3

Use your preferred backup method to make a complete (level-0) backup of your database server. You can use ON-Bar, **ontape**, or ON-Archive to perform a backup.

To start a backup of the database server on UNIX, you can enter **SINFORMIXDIR/bin/bar** where the database server is installed. ♦

On Windows NT, you can double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group.

Windows NT does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup.

For more information on how to make backups, refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.

Export Data from Dynamic Server 7.3

If you are migrating to a different environment (for example, to UNIX from Windows NT), choose one of the following sets of migration utilities:

- **dbexport** and **dbimport** (see [“Using the dbexport and dbimport Utilities” on page 16-8](#))
- **UNLOAD**, **dbschema**, and **LOAD** (see [“Using UNLOAD, dbschema, and LOAD” on page 16-9](#))
- **UNLOAD**, **dbschema**, and **dbload** (see [“Using UNLOAD, dbschema, and dbload” on page 16-10](#))

Using the dbexport and dbimport Utilities

If you intend to move an entire database to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3 in different environments, the **dbexport** and **dbimport** combination is the easiest migration method.

Perform the following steps:

1. Use **dbexport** to export the data from Dynamic Server 7.3.
You can move the data to a directory or directly to tape. Do not use the **-ss** option when you move data between database servers.
2. Remove table-fragmentation expressions from the CREATE TABLE statements in the schema file (the **.sql** file that **dbexport** creates).
If tables are fragmented, **dbimport** might not work. Informix recommends that you defragment the tables before you export the database.
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges for specific tables and indexes
 - Change the logging status of the database
 - Change the ANSI-compliance status of the database
 - Remove unsupported SQL syntax, if necessary
4. Follow the instructions in your *Installation Guide* and your *Administrator's Guide* to install and configure Dynamic Server, Workgroup and Developer Editions.
5. Change the **INFORMIXSERVER** environment variable to specify your target database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Move to the directory where **dbimport** will store the Dynamic Server, Workgroup and Developer Editions, database.
8. Use **dbimport** to move the data to a Dynamic Server, Workgroup and Developer Editions, database.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 20, "Using the dbexport and dbimport Utilities."](#)

Using the **UNLOAD** Statement

The **UNLOAD** statement lets you retrieve selected rows from a database and write those rows to a text file.

Using **UNLOAD**, **dbschema**, and **LOAD**

If you want to move selected tables or columns instead of an entire Dynamic Server 7.3 database to Dynamic Server, Workgroup and Developer Editions, use the **UNLOAD** and **LOAD** statements in the DB-Access utility with the **dbschema** utility.

To use **UNLOAD, **dbschema**, and **LOAD** to move data to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3**

1. Make sure that you have sufficient disk space to store the unloaded data. (The **UNLOAD** statement does not allow you to unload to tape.)
2. Invoke the DB-Access utility.
3. To move the selected data into text files, use **UNLOAD** statements. Use a separate **UNLOAD** statement for each target table.
4. Exit DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. To create a schema file from the source database server, use the **dbschema** utility.
 - b. Edit the schema file so that it describes the new tables.
If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.
6. Follow the instructions in your *Installation Guide* and your *Administrator's Guide* to install and configure Dynamic Server, Workgroup and Developer Editions.
7. Change the **INFORMIXSERVER** environment variable to specify your target database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.

10. Select the target database.
11. If you are creating a new database, execute the CREATE DATABASE statement or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5 or enter CREATE TABLE statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. To load the data into the desired tables, use LOAD statements.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data in the specified UNLOAD file before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move data to Dynamic Server, Workgroup and Developer Editions, from Dynamic Server 7.3

1. Follow steps 1 through 13 from [“Using UNLOAD, dbschema, and LOAD” on page 16-9](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to Section VIII, [“Data Migration Utilities.”](#) For information on how to use DB-Access, refer to the *DB-Access User’s Manual*.

Shut Down Dynamic Server 7.3

Communicate to client users how long you expect the database server to be off-line for the migration. Terminate all database server processes and place your database server in quiescent mode (also called *administration mode*).

To shut down the database server gracefully

1. Warn all users that you plan to shut down the database server and wait for them to exit.
2. Become user **informix** on UNIX. On Windows NT, you must be a member of the **Informix-Admin** group.

Use **Services** in the Windows NT **Control Panel**. ♦

3. Execute the following command to take the database server to quiescent mode:

```
onmode -sy
```

4. Wait until your database server is in quiescent mode.

To verify the mode of your database server, execute the **onstat** - command. The first line of the **onstat** output contains the status of your database server. [Figure 16-2](#) shows that the database server is in quiescent mode.

```
Informix OnLine Dynamic Server Version x.xx.xxx -- Quiescent -- Up xx:xx:xx -- xxxx
```

Informix Dynamic Server is in quiescent mode.

Kbytes

Figure 16-2
Example of **onstat**
Status Line

5. Execute the following command to force a new logical log:
6. Execute the following command to force a checkpoint:
7. Execute the following command to shut down the database server:

```
onmode -l
```

```
onmode -c
```

```
onmode -yuk
```

Tip: Monitor your log activity to verify that all commands were executed properly and to check for inconsistencies prior to migration.



WIN NT

UNIX

Install and Configure Dynamic Server, Workgroup and Developer Editions

If you have not already done so, follow the instructions in your *Installation Guide* to install and configure your target database server.

You can install Informix Enterprise Command Center (IECC) on a personal computer that runs Windows 95 or Windows NT, and the database server on a UNIX computer. The installation program also starts the server agent, which is the communication link between Dynamic Server, Workgroup and Developer Editions, and the IECC client. ♦

WIN NT

You can install Dynamic Server, Workgroup and Developer Editions, and IECC on either the same or different computers. ♦

Use the **Setup** program to specify the network protocol and the computer on which Dynamic Server, Workgroup and Developer Editions, looks for the database server definitions (**sqlhosts** and **osahosts** definitions).

Verify Port Numbers and the Services File

The **services** file contains service names, port numbers, and protocol information. If you have installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and on the server computers.

UNIX

For UNIX operating systems not running NIS, the **services** file resides in the **/etc/services** directory on the database server and in the **\windir\services** directory on the Windows 95 client. ♦

WIN NT

The **services** file resides in the **\windir\system32\drivers\etc** directory. ♦

Customize the Database Server Environment

If you are an advanced user, you can customize the **ONCONFIG** configuration file and environment variables for Dynamic Server, Workgroup and Developer Editions. Use a text editor to edit the **ONCONFIG** file. For more information on configuration parameters, refer to your *Installation Guide*.



Important: Use the same values for your target database server for `ROOTOFFSET`, `ROOTSIZE`, and `ROOTPATH` that you used for your source database server.

You might want to customize new environment variables on the client. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Bring Dynamic Server, Workgroup and Developer Editions, On-Line

UNIX

To start your database server at the UNIX command line, enter `oninit` on the server computer. ♦

If you customized the database server environment, you can bring down and restart your database server with IECC. When you restart Dynamic Server, Workgroup and Developer Editions, the changes to the configuration parameters and environment variables take effect.

WIN NT

To start the database server with IECC on Windows NT

1. In the **Informix Administration Tools** program group, double-click the **Command Center** icon.
2. In IECC, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to the 7.3 IECC manual. ♦

Import Data into Dynamic Server, Workgroup and Developer Editions

To load data into Dynamic Server, Workgroup and Developer Editions, use **dbimport**, **LOAD**, or **dbload**, depending on which utility you used to export the databases.

UNIX

WIN NT

Verify the Integrity of the Data

To verify the integrity of data, use the **oncheck** utility, as described in your *Administrator's Guide*.

Make an Initial Backup of Dynamic Server, Workgroup and Developer Editions

Use your preferred backup method to make a complete (level-0) backup of your database server. You can use ON-Bar, **ontape**, or ON-Archive to perform a backup.

To start a backup of the database server on UNIX, you can enter **\$INFORMIXDIR/bin/bar** where the database server is installed. ♦

On Windows NT, you can double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group.

Windows NT does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup.

For more information on how to make backups, refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.

Run UPDATE STATISTICS

After you complete the migration procedure, run the UPDATE STATISTICS statement on the database server. The UPDATE STATISTICS statement updates the information that Dynamic Server, Workgroup and Developer Editions, uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Complete Migration

The first time your target database server is brought on-line, the **sysmaster** and the **sysutils** databases are built. Check the message log to ensure that the **sysutils** and **sysmaster** databases are created successfully before you allow users to access the database server. After you ensure that client users can access data on your target database server, the migration process is complete.

Adapt Your Programs for Dynamic Server, Workgroup and Developer Editions

After you successfully migrate the database server data, verify that your application developers know the differences between the source and target database servers. Dynamic Server, Workgroup and Developer Editions, supports the same products and features as Dynamic Server 7.3, except for the High-Performance Loader (HPL) and the following features:

- Fragmentation (also known as partitioning)
- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that Dynamic Server, Workgroup and Developer Editions, supports, refer to the *Informix Guide to SQL: Syntax* and the *Informix Guide to SQL: Reference*.

Migrating Data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on a Different Operating System

The following sections describe the steps for moving data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, on a different operating system.

When you move data between UNIX and Windows NT, you must choose a migration utility, eliminate database server- and environment-specific features, migrate the data, and modify applications.

Complete the following migration steps:

1. Save copies of the current configuration files.
2. Verify the integrity of the data.
3. Make a final backup of Dynamic Server, Workgroup and Developer Editions.
4. Export data from Dynamic Server, Workgroup and Developer Editions.
5. Bring Dynamic Server, Workgroup and Developer Editions, off-line.
6. Install and configure Dynamic Server 7.3.
7. Verify port numbers and the **services** file.
8. Customize the database server environment (optional).
9. Bring Dynamic Server 7.3 on-line.
10. Import data into Dynamic Server 7.3.
11. Verify the integrity of the data.
12. Make an initial backup of Dynamic Server 7.3.
13. Run UPDATE STATISTICS.
14. Complete migration.
15. Adapt your programs for Dynamic Server 7.3.

Save Copies of the Current Configuration Files

Save copies of the current configuration files that you have modified, as follows:

- Current **ONCONFIG** file, located in the **etc** subdirectory of your installation directory
- **sqlhosts** information
- **adtcfg**, located in the **aaodir** subdirectory
- **adtmasks.***, located in the **dbssodir** subdirectory
- ON-Archive configuration files, located in the **etc** subdirectory

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in your *Administrator's Guide*.

Back Up Dynamic Server, Workgroup and Developer Editions

Use your preferred backup method to make a complete (level-0) backup of your database server. You can use ON-Bar, **ontape**, or ON-Archive to perform a backup.

UNIX

To start a backup of the database server on UNIX, you can enter **SINFORMIXDIR/bin/bar** where the database server is installed. ♦

WIN NT

On Windows NT, you can double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group.

Windows NT does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup.

For more information on how to make backups, refer to your *Backup and Restore Guide* or the *Archive and Backup Guide*.

Export Data from Dynamic Server, Workgroup and Developer Editions

To migrate to a different operating system (such as to Windows NT from UNIX), choose one of the following sets of data migration utilities:

- **dbexport** and **dbimport** (see [“Using the dbexport and dbimport Utilities” on page 16-18](#))
- **UNLOAD**, **dbschema**, and **LOAD** (see [“Using UNLOAD, LOAD, and dbschema” on page 16-19](#))
- **UNLOAD**, **dbschema**, and **dbload** (see [“Using UNLOAD, dbschema, and dbload” on page 16-20](#))

UNIX

Using the dbexport and dbimport Utilities

If you intend to move an entire database to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions, in different environments, the **dbexport** and **dbimport** combination is the easiest migration method. Follow these steps:

1. Use **dbexport** to export the data from the source database server.
You can move the data to a directory or directly to tape. Do not use the **-ss** option when you move data between database servers.
2. You might want to add the following information that Dynamic Server 7.3 databases and tables can use:
 - Fragmentation schemes
 - PDQ support ♦
3. You can also make the following changes in the schema file:
 - Alter ownership or SQL privileges for specific tables and indexes.
 - Change the logging status of the database.
 - Change the ANSI-compliance status of the database.
 - Remove unsupported SQL syntax, if necessary.
4. Follow the instructions to install and configure Dynamic Server 7.3.
5. Change the **INFORMIXSERVER** environment variable to specify your new database server.
6. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
7. Move to the directory where **dbimport** will store the Dynamic Server 7.3 database.
8. Use **dbimport** to move the data to a Dynamic Server 7.3 database.

For detailed descriptions of the **dbexport** and **dbimport** utilities, refer to [Chapter 20, “Using the dbexport and dbimport Utilities.”](#)

Using the UNLOAD Statement

The UNLOAD statement lets you retrieve selected rows from a database and write those rows to a text file.

Using UNLOAD, LOAD, and dbschema

If you want to move selected tables or columns instead of an entire Dynamic Server, Workgroup and Developer Editions, database to Dynamic Server 7.3, use the UNLOAD and LOAD statements in DB-Access with the **dbschema** utility.

To use UNLOAD, LOAD, and dbschema to move data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions

1. Make sure that you have sufficient disk space to store the unloaded data. (The UNLOAD statement does not allow you to unload to tape.)
2. Invoke DB-Access.
3. Use UNLOAD statements to move the selected data into text files. Use a separate UNLOAD statement for each target table.
4. Exit from DB-Access.
5. If you plan to load data into a table or tables that do not exist, follow these steps:
 - a. Use the **dbschema** utility to create a schema file from the Dynamic Server 7.3 database.
 - b. Edit the schema file so that it describes the new tables.If you prefer, you can omit this step and, in step 12, enter the statements that create the tables.
6. Follow the instructions to install and configure Dynamic Server 7.3.
7. Change the **INFORMIXSERVER** environment variable to specify your new database server.
8. If necessary, use your standard file-transfer process to move the export files to the target computer. Ensure that both database servers use the same **DBDATE** and **DBMONEY** formats.
9. Invoke the DB-Access utility.
10. Select the target database.

11. If you are creating a new database, execute the CREATE DATABASE statement or choose **Database→Create** from the DB-Access menu.
12. If you plan to load data into a new table, choose and run the schema file that you prepared in step 5 or enter CREATE TABLE statements to create the new tables.
13. If you created a new database, you might need to update the **DBPATH** environment variable with the new database location.
14. Use LOAD statements to load the data into the desired tables.

Using UNLOAD, dbschema, and dbload

If you need to manipulate the data in the specified UNLOAD file before you load it into a new table, use a combination of the UNLOAD statement and the **dbschema** and **dbload** utilities.

To use UNLOAD, dbschema, and dbload to move data to Dynamic Server 7.3 from Dynamic Server, Workgroup and Developer Editions

1. Follow steps 1 through 13 from [“Using UNLOAD, LOAD, and dbschema” on page 16-19](#).
2. Build a command file to use with the **dbload** utility, which loads the data files into the tables.
3. Execute **dbload** to load the data as your command file directs.

For information on UNLOAD, LOAD, **dbload**, and **dbschema**, refer to Section VIII, [“Data Migration Utilities.”](#) For information on how to use DB-Access, refer to the *DB-Access User’s Manual*.

Shut Down Dynamic Server, Workgroup and Developer Editions

UNIX

WIN NT

Select **Off-Line** from the **Server Mode** list box in the **General** tab of IECC. ♦

In IECC, select the database server in the **All Servers** tree view or the **Servers** list box. Choose **Server→Off-line**. ♦

Install and Configure Dynamic Server 7.3

Install and configure Dynamic Server 7.3 according to the instructions in your *Installation Guide*.

UNIX

You must be user **root** to install the product. When you finish the installation and system reconfiguration, exit as user **root** and log in as user **informix**. ♦

WIN NT

You can install the target database server and the administration tools on either the same or different computers.

The installation program automatically starts the target database server.

Use **Setup** to specify the network protocol and the computer on which the target database server looks for the database server definitions. ♦

Verify Port Numbers and the Services File

The **services** file contains service names, port numbers, and protocol information. If you have installed the database server and the administration tools on different computers, verify that the port number listed in the **services** file is the same on the client and the server computers.

UNIX

The **services** file resides in the **/etc/services** directory on the server and in the **\windir\services** directory on the Windows NT client. ♦

WIN NT

The **services** file resides in the **\windir\system32\drivers\etc** directory. ♦

Customize the Database Server Environment

If you are an advanced user, you can customize the ONCONFIG configuration file and environment variables for Dynamic Server 7.3. To edit the ONCONFIG file, use a text editor. For more information on configuration parameters, refer to your *Administrator's Guide*. For more information on environment variables, refer to the *Informix Guide to SQL: Reference*.

Important: Use the same values for your target database server for **ROOTOFFSET**, **ROOTSIZE**, and **ROOTPATH** that you used for your source database server.



UNIX

Bring Dynamic Server 7.3 On-Line

The installation program brings Dynamic Server 7.3 on-line automatically.

If you customized the database server environment, you can bring down and restart the target database server with IECC. When you restart the target database server, the changes to the configuration parameters and environment variables take effect.

To start Dynamic Server 7.3 on UNIX

1. Enter **oninit** at the command-line prompt.
2. If you want to initialize the database server, enter **oninit -i**. ♦

To start Dynamic Server 7.3 on Windows NT

1. In the **Informix Administration Tools** program group, double-click the IECC icon.
2. In IECC, select the database server in the **All Servers** tree view or the **Servers** list box.
3. Choose **Server→On-line**.

For more information, refer to the 7.3 IECC manual. ♦

Import the Data into the Target Database Server

Use **dbimport**, **LOAD**, or **dbload** to load the data into Dynamic Server 7.3, depending on which utility you used to export the data.

Verify the Integrity of the Data

Use the **oncheck** utility to verify the integrity of data, as described in your *Administrator's Guide*.

WIN NT

UNIX

WIN NT

Back Up the Target Database Server

Use your preferred backup method to make a complete (level-0) backup of your database server. You can use ON-Bar, **ontape**, or ON-Archive to perform a backup.

To start a backup of the database server on UNIX, you can enter **SINFORMIXDIR/bin/bar** where the database server is installed. ♦

On Windows NT, you can double-click the **Backup and Restore** icon in the **Informix Administration Tools** program group.

Windows NT does not support ON-Archive. ♦

The tape parameters must specify a valid tape device. Be sure to retain and properly label the tape volume that contains the backup.

For more information on how to make backups, refer to your *Backup and Restore Guide* or *Archive and Backup Guide*.

Run UPDATE STATISTICS

After you complete the migration procedure, run the UPDATE STATISTICS statement on the server. The UPDATE STATISTICS statement updates the information that Dynamic Server 7.3 uses to plan efficient queries. For more information about UPDATE STATISTICS, refer to the *Informix Guide to SQL: Syntax*.

Complete Migration

The first time the target database server is brought on-line, the **sysmaster** and **sysutils** databases are built. Check the message log to ensure that the database builds are complete before you allow users to access the database server. After you ensure that client users can access data on your target database server, the migration process is complete.

Adapt Your Programs for Dynamic Server 7.3

After you successfully move the data to Dynamic Server 7.3, verify that your application developers know the differences between both database servers. Dynamic Server 7.3 supports the same products and features as Dynamic Server, Workgroup and Developer Editions, plus the High-Performance Loader (HPL) and the following features:

- Fragmentation (also known as partitioning)
- Parallel data query (PDQ)
- Role separation

For information on the SQL statements that Dynamic Server 7.3 supports, refer to the *Informix Guide to SQL: Syntax* and the *Informix Guide to SQL: Reference*.

Locale Changes

- Chapter 17** **Understanding Language Support in Informix Database Servers**
- Chapter 18** **Converting to GLS**
- Chapter 19** **Reverting from GLS**



Understanding Language Support in Informix Database Servers

In This Chapter	17-3
Informix Language Support	17-4
Global Language Support	17-4
Informix GLS Products	17-5
GLS Locales	17-6
GLS Environment Variables	17-8
Code-Set Conversion	17-8
Native Language Support	17-9
Informix NLS Products	17-10
NLS Locales	17-11
NLS Environment Variables	17-12
NLS Support by Informix GLS Products	17-13
Asian Language Support	17-14
Informix ALS Products	17-15
ALS Locales	17-15
ALS Environment Variables	17-16
ALS Support by Informix GLS Products	17-17
Language Support in Database Servers	17-20
Dynamic Server 2000 and Universal Server	17-21
Extended Parallel Server, Dynamic Server AD/XP, and OnLine XPS	17-22
Dynamic Server 7.3 and OnLine Dynamic Server	17-22
INFORMIX-SE	17-23
Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server	17-24

ALS

NLS

GLS

In This Chapter

Informix products use the following language-support features:

- Informix Asian Language Support (ALS) provides support for Asian (multibyte) characters. ♦
- Informix Native Language Support (NLS) provides support for single-byte, non-English characters. ♦
- Informix Global Language Support (GLS) provides support for both multibyte and single-byte characters. ♦

This chapter describes how to migrate a database that uses the NLS or ALS feature for its language support to the new GLS support. The chapter discusses the following topics:

- Informix language support
- Language support in database servers

For an introduction to the GLS feature, see the *Informix Guide to GLS Functionality*.

Informix Language Support

By default, most Informix products assume that a database contains character data that follows the conventions of U.S. English. These conventions include the following:

- Character data that uses a code set that supports U.S. English characters (the ASCII code set)
- Classification of character data into English categories
- Collation of character data in ASCII code-set order

A *non-English database* is a database whose character data is in some language other than English. Informix has provided the ALS language support for non-English databases with Asian (multibyte) characters and NLS language support for non-English databases with single-byte characters. With Version 7.2, Informix created a single feature to provide support for single-byte and multibyte data in non-English languages. This feature is called Informix Global Language Support (GLS). The following sections summarize each of the Informix language support features.

GLS

Global Language Support

The GLS feature provides support for both single-byte and multibyte non-English data in a database application. This feature provides the following advantages to a database application:

- Truly internationalized support for languages
The database server that uses the GLS feature can support data in both single-byte and multibyte code sets. You do not need to install two different products.
- Informix-specific locale files
The *GLS locale* is not operating-system dependent. Your database, database server, and client applications are no longer dependent on the language support that your operating system provides.

This section provides the following information about the GLS feature:

- Which Informix products support the GLS feature?
- What is a GLS locale?
- What are the GLS environment variables?
- How do client applications perform code-set conversion with a GLS database server?

For a more complete description of the GLS feature, see the *Informix Guide to GLS Functionality*.

Informix GLS Products

Informix GLS products use the GLS feature to provide support for non-English database applications. Informix supports the GLS feature in the following types of products:

- Database server products
- Client products

Informix introduced GLS in its Version 7.2 database servers.

GLS Database Servers

A database server that uses the GLS feature for its language support is called a *GLS database server*. A database server uses the GLS feature to correctly manipulate character, numeric, and date quantities, as well as to sort character data. As [Figure 17-8 on page 17-21](#) shows, the following database servers are GLS database servers:

- Dynamic Server 2000
- Universal Server
- Extended Parallel Server
- Dynamic Server with AD and XP Options
- Dynamic Server, Version 7.3x
- Dynamic Server, Workgroup and Developer Editions, Version 7.3x
- OnLine Dynamic Server, Version 7.2x
- OnLine Workgroup Server, Version 7.2x
- SE, Version 7.2x

Informix GLS database servers create *GLS databases*. If you do not set the **CLIENT_LOCALE** and **DB_LOCALE** environment variables, a GLS database server creates an English database (a database with the default locale, U.S. English). For more information on these GLS environment variables, see [“GLS Environment Variables” on page 17-8](#).

GLS Client Products

A client product (such as Informix ESQL/C) that uses the GLS feature for its language support is called a *GLS client product*. A client application uses the GLS feature to provide end-user formats for date, time, monetary, and numeric values.

Informix GLS includes the following client products:

- Version 9.x ESQL/C
- Version 8.2 ESQL/C
- Version 7.2x ESQL/C and ESQL/COBOL
- Version 7.1TD1 ESQL/C and ESQL/COBOL for Microsoft Windows Environments
- Version 5.x (WC1 and later) ESQL/C for Windows

GLS Locales

A GLS database server uses a GLS locale to provide language support for a database. A *GLS locale* is a set of Informix files that bring together the information about data that is specific to a particular culture, language, or territory. In particular, a GLS locale provides the following information:

- The name of the code set that the application data uses
- The collation order to use for character data
- The format for different types of data to appear to end users

A GLS locale file groups locale-specific information into the following locale categories.

Locale Category	Description
CTYPE	Controls the behavior of character classification and case conversion.
COLLATION	Controls the behavior of string comparisons.
NUMERIC	Controls the behavior of non-monetary numeric end-user formats.
MONETARY	Controls the behavior of currency end-user formats.
TIME	Controls the behavior of date and time end-user formats.
MESSAGES	Controls the definitions of affirmative and negative responses to messages.

For GLS databases, the database server stores a condensed version of the database locale in the following two rows of the **systables** system catalog table:

- The row with **tabid** 90 stores the COLLATION category of the database locale.
The COLLATION category of a locale determines the order in which the characters of the code set collate. The database server uses the COLLATION category of the database locale to collate character data. The **tablename** value for this row is GLS_COLLATE.
- The row with **tabid** 91 stores the CTYPE category of the database locale.
The CTYPE category of a locale determines how characters of the code set are classified. The database server uses character classification for case conversion and some regular-expression evaluation. The **tablename** value for this row is GLS_CTYPE.

The rows with the values 90 and 91 in the **tabid** column of the **systables** system catalog table store the condensed locale name in the **site** column.

GLS Environment Variables

Figure 17-1 shows the locale environment variables that GLS products support.

Figure 17-1
GLS Environment Variables

GLS Environment Variable	Purpose
CC8BITLEVEL	Specifies how the C compiler handles multibyte characters.
CLIENT_LOCALE	Specifies the name of the client locale.
DB_LOCALE	Specifies the name of the database locale.
ESQLMF	Specifies whether to invoke the ESQL/C multibyte filter, esqlmf .
GL_DATE	Supports extended format strings for international formatting of DATE values.
GL_DATETIME	Supports extended format strings for international formatting of DATETIME values.
GLS8BITFSYS	Specifies how to handle filenames that contain non-ASCII characters.
SERVER_LOCALE	Specifies the name of the database server locale.

For backward compatibility, GLS products also support all the NLS environment variables (see Figure 17-3 on page 17-12) and a subset of the ALS environment variables (see Figure 17-5 on page 17-16).

Code-Set Conversion

A client application performs code-set conversion when the client locale does not match the database locale. In Figure 17-2, the **Code-Set Conversion** column shows the environment variables that the NLS and ALS client products use to perform code-set conversion. The **Client Locale** and **Database Locale** columns show the environment variables from the client application that the GLS database server uses to set the client and database locale, respectively.

Figure 17-2
Locale Information Sent to GLS Database Servers

Client Product	GLS Database Server		
	Client Locale	Database Locale	Code-Set Conversion
GLS client products: (For a list of products, see “GLS Client Products” on page 17-6.)	CLIENT_LOCALE	DB_LOCALE	Between CLIENT_LOCALE and DB_LOCALE
NLS client products: 8.1x ESQL/C 6.x through 7.1x ESQL/C and ESQL/COBOL	LANG, LC_* (with DBNLS=1)	LANG, LC_* (with DBNLS=1)	Between DBAPICODE and LANG, LC_* (WITH DBNLS=1)
7.1TC1 ESQL/C for Win32 5.X (PRE-WG1) ESQL/C for Windows	LANG, LC_* (with DBNLS=1)	LANG, LC_* (with DBNLS=1)	Between CLIENT_LOCALE and DB_LOCALE
ALS client products: Version 6.x ALS ESQL/C	CLIENT_LOCALE	DB_LOCALE	Between CLIENT_LOCALE and DB_LOCALE
Version 5.x ALS ESQL/C	DBCODESET	DBCODESET	Between DBAPICODE and DBCODESET



Tip: In [Figure 17-2](#), LC_* is an abbreviation for the following NLS environment variables: LC_CTYPE, LC_COLLATE, LC_MONETARY, LC_NUMERIC and LC_TIME. For more information, see “[NLS Environment Variables](#)” on page 17-12.

For more information on code-set conversion, see the *Informix Guide to GLS Functionality*.

NLS

Native Language Support

The NLS feature provides support for single-byte, non-English data in a database application. These code points are the ASCII code set. If a code set contains more than 128 characters, some of its characters are 8-bit characters, which indicates that the eighth bit of the byte is set. Both ASCII and 8-bit characters are single-byte characters.



Tip: NLS products do not support multibyte code sets. Previously, users of multibyte code sets relied on locally customized versions of the products such as the Informix ALS products. For more information, see [“Asian Language Support” on page 17-14](#).

This section provides the following information about the NLS feature:

- Which Informix products support the NLS feature?
- What is an NLS locale?
- What are the NLS environment variables?
- How do Informix GLS products support NLS products?

For more information about how to use NLS products, refer to the *Informix Guide to SQL: Reference*, Version 7.1; the *INFORMIX-ESQL/C Programmer's Manual*, Version 6.0; and the *INFORMIX-ESQL/COBOL Programmer's Manual*, Version 6.0.

Informix NLS Products

Informix NLS products are those that use the NLS feature to provide support for non-English database applications that use single-byte code sets. Informix supports the NLS feature in the following types of products:

- Database server products
- Client products

Informix introduced Native Language Support (NLS) in its Version 6.0 products.

NLS Database Servers

A database server that uses the NLS feature for its language support is called an *NLS database server*. A database server uses the NLS feature to provide the code set for data, collation order, and character classification. The following database servers are NLS database servers:

- OnLine Dynamic Server, Version 6.0 through Version 7.1x
- SE, Version 6.0 through Version 7.1x
- OnLine Workgroup Server, Version 7.12

Informix NLS database servers create *NLS databases*. If you do not set the **LANG** and **DBNLS** environment variables, an NLS database server creates an English database (one that uses the default operating-system locale). For more information on these NLS environment variables, see [“NLS Environment Variables” on page 17-12](#).

NLS Client Products

A client product (such as Informix ESQL/C) that uses the NLS feature for its language support is called an *NLS client product*. A client application uses the NLS feature to provide end-user formats for date, time, monetary, and numeric values.

Informix NLS client products include the following list:

- Version 7.1x ESQL/C and ESQL/COBOL
- Version 6.x ESQL/C and ESQL/COBOL
- Version 7.1TC1 ESQL/C and ESQL/COBOL for Microsoft Windows Environments
- Version 5.x (pre-WG1) ESQL/C for Windows

NLS Locales

An NLS product uses locales that are native to the operating system to provide language support. These locales are called *NLS locales (or operating-system locales)*. An NLS locale groups locale-specific information into the following locale categories.

Locale Category	Description
CTYPE	Controls the behavior of character classification and case conversion.
COLLATION	Controls the behavior of string comparisons.
NUMERIC	Controls the behavior of non-monetary numeric end-user formats.
MONETARY	Controls the behavior of currency end-user formats.
TIME	Controls the behavior of date and time end-user formats.

For NLS databases, the database server stores the name of the database locale in the following two rows of the **systables** system catalog table:

- The row with **tabid** 90 stores the COLLATION category of the database locale.
The COLLATION category of a locale determines the order in which the characters of the code set collate. The database server uses the COLLATION category of the database locale to collate character data. The **tablename** value for this row is NLSCOLL.
- The row with **tabid** 91 stores the CTYPE category of the database locale.
The CTYPE category of a locale determines how characters of the code set are classified. The database server uses character classification for case conversion and some regular-expression evaluation. The **tablename** value for this row is NLSCTYPE.

The rows with the values 90 and 91 in the **tabid** column of the **systables** system catalog table store the condensed locale name in the **site** column.

NLS Environment Variables

An NLS product uses both operating-system environment variables and Informix NLS environment variables. [Figure 17-3](#) lists Informix NLS environment variables.

Figure 17-3
NLS Environment Variables

NLS Environment Variable	Purpose
COLLCHAR	Allows client applications to use NLS collation.
DBAPICODE	Allows client applications to use a different code set than the one that the database server uses.
DBDATE	Specifies an end-user format for values in DATE columns.
DBLANG	Specifies the location of product-specific message files.
DBMONEY	Specifies an end-user format for values in MONEY columns.
DBNLS	Enables NLS features.

(1 of 2)

NLS Environment Variable	Purpose
DBTIME	Specifies an end-user format for values in DATETIME columns.
LANG	Specifies the operating-system locale for your NLS features.
LC_COLLATE	Specifies a collating sequence for your locale-sensitive data.
LC_CTYPE	Affects the behavior of regular expressions and character-evaluation functions.
LC_MONETARY	Specifies the format and national currency symbol for monetary values.
LC_NUMERIC	Specifies the format and decimal separator for numeric values.
LC_TIME	Specifies the format for national dates and times.

(2 of 2)

NLS Support by Informix GLS Products

Informix GLS products support all NLS environment variables. Therefore, Informix products can provide the following support for NLS products:

- An NLS client application can access a GLS database server.
The GLS database server receives the NLS environment variables and uses them to determine client and database locale names (**CLIENT_LOCALE** and **DB_LOCALE**). It looks for NLS-compliant GLS locales that have the same name. The client application performs any code-set conversion between the **DBAPICODE** and **LANG** environment variables. For more information, see [Figure 17-2 on page 17-9](#).
- A GLS client application can access an NLS database server with minimal modification.
If the **DB_LOCALE** environment variable is set, the GLS client application performs code-set conversion between the **LC_CTYPE** (or **LANG**, if **LC_CTYPE** is not set) and **DB_LOCALE** locales. If **CLIENT_LOCALE** is also set, the GLS client application performs code-set conversion between the **CLIENT_LOCALE** and **DB_LOCALE** locales.

To use the NLS feature, you *must* set the following environment variables in the client environment:

- Set the **DBNLS** environment variable to 1 or 2 to enable the NLS feature.
- Set the **LANG** environment variable to the name of the NLS locale that the NLS database uses.

The remaining NLS environment variables in [Figure 17-3 on page 17-12](#) can override some or all of the basic NLS functionality that **DBNLS** and **LANG** determine. The *Informix Guide to GLS Functionality* discusses the precedence of NLS and GLS environment variables.

If you have installed a GLS database server and want your NLS database to work with this new database server, you must upgrade the NLS database to a GLS database. For more information, see [“Upgrading to GLS from NLS” on page 18-6](#).

ALS

Asian Language Support

The ALS feature provides support for multibyte Asian data in a database application. If a character set contains more than 256 characters, the code set must contain multibyte characters. A *multibyte character* might require from 2 to 4 bytes of storage.



***Tip:** ALS products support only multibyte code sets for several Asian languages. Previously, users of single-byte code sets relied on the Informix NLS products. For more information, see [“Native Language Support” on page 17-9](#).*

This section provides the following information about the ALS feature:

- Which Informix products support the ALS feature?
- What is an ALS locale?
- What are the ALS environment variables?
- How do Informix GLS products support ALS products?

For more information about how to use ALS products, refer to your Informix ALS documentation.

Informix ALS Products

Informix ALS products use the ALS feature to provide support for non-English database applications that use Asian multibyte code sets. Before Version 7.2x, the following locally customized versions of Informix products supported Asian languages that use multibyte characters:

- Version 4.x ASCII Japanese (4.s)
- Informix ALS products, Versions 4.x, 5.x, and 6.x

An OnLine or SE database server that uses the ALS feature for its language support is called an *ALS database server*. Informix ALS database servers create *ALS databases*. A client product that uses the ALS feature for its language support is called an *ALS client product*.

ALS Locales

An ALS product includes Asian locales for Asian-language support. For ALS databases, the database server stores a condensed version of the database locale in the **systables** system catalog table. The location of this information depends on the version of the Informix ALS product, as [Figure 17-4](#) shows.

Figure 17-4
Location in systables of ALS Locale Information

ALS Version Number	Location in systables
Version 6.x	In the rows whose tabid values are 95 and 96
Version 5.x	In the rows whose tabid values are 95, 96, and 97
Version 4.x	In a row whose tabid value is 98
Version 4.x ASCII	None (locale is always Japanese with the SJIS code set)

ALS Environment Variables

Figure 17-5 lists Informix ALS environment variables.

Figure 17-5
ALS Environment Variables

ALS Environment Variable	Purpose	Supported by GLS Products?
ALS8BITFSYS	Enables ALS products to handle 8-bit filenames.	No (use GLS8BITFSYS instead)
CC8BITLEVEL	Specifies the preprocessing format for the ESQL/C compiler.	Yes
CLIENT_LOCALE	For Version 5.x ALS, specifies the code set that the client application uses. For Version 6.x ALS, specifies the locale that the client application uses.	Yes
DBCDESET	Similar to CLIENT_LOCALE and DB_LOCALE; used by Version 5.x ALS.	No
DBCSOEVERIDE	Forces DB_LOCALE values to override default restrictions on how to access databases.	No
DBDATE	Specifies an end-user format for values in DATE columns.	Yes (use GL_DATE instead)
DBFORMAT	Specifies formats for INFORMIX-SQL and INFORMIX-4GL.	No
DBINFXRC	Specifies the pathname of a user-defined environment-variable configuration file.	No
DBLANG	Specifies the location of product-specific message files.	Yes

(1 of 2)

ALS Environment Variable	Purpose	Supported by GLS Products?
DB_LOCALE	For Version 5.x ALS, specifies the code set that locale-sensitive data in the database uses. For Version 6.x ALS, specifies the locale of the database. You must set the DB_LOCALE environment variable to store and access multibyte Asian characters in an ALS database. Otherwise, the database server assumes the locale to be U.S. English (en_US.8859-1).	Yes
DBMONEY	Specifies an end-user format for values in MONEY columns.	Yes
DBTIME	Specifies an end-user format for values in DATETIME columns.	Yes

(2 of 2)

ALS Support by Informix GLS Products

Informix GLS client applications support most ALS environment variables. (For more information, see [Figure 17-5 on page 17-16](#).) Therefore, Informix GLS products provide the following support for ALS products:

- An ALS client application can access a GLS database server.
For information on how a GLS database server supports ALS client applications, see [Figure 17-6](#).
- A GLS client application can access an ALS database server.
For information on how GLS client applications support connections to ALS database servers, see [Figure 17-7 on page 17-19](#).

Figure 17-6

GLS Database Server Support for ALS Client Applications

ALS Client Application	Support in GLS Database Server
Version 6.x ALS ESQL/C applications	<p>The database server obtains the database locale name from the DB_LOCALE environment variable.</p> <p>The client application performs code-set conversion between the locales that CLIENT_LOCALE and DB_LOCALE specify.</p>
Version 5.x ALS ESQL/C applications	<p>The database server obtains the database locale name from the DBCODESET environment variable. It converts this value to a valid GLS locale name and uses this locale name for DB_LOCALE.</p> <p>The client application performs code-set conversion between the locales that DBAPICODE and DBCODESET specify.</p>
Version 4.x ALS ESQL/C applications, Version 4.x ALS Viewpoint	<p>The database server obtains the database locale name from the DBCODESET environment variable. It converts this value to a valid GLS locale name and uses this locale name for DB_LOCALE.</p> <p>The client application cannot perform code-set conversion.</p>
Version 4.x ASCII ESQL/C applications	<p>The database server obtains the database locale name from the DB_LOCALE environment variable <i>on the server computer</i>. This environment variable must be set to the Japanese locale, ja_jp.sjis, <i>before</i> the database server is initialized.</p> <p>The client application does not need to perform code-set conversion.</p>

Figure 17-7
GLS Client Application Support for ALS Database Servers

ALS Database Server	Support in GLS Client Application
Version 6.x ALS database server	You must set the DB_LOCALE environment variable to the name of the database locale for the ALS database.
Version 5.x ALS database server	You must set the CLIENT_LOCALE and DB_LOCALE environment variables if the GLS client application is to perform code-set conversion. The DBCODESET environment variable enables the connection. This environment does not affect the client application but is sent on to the ALS database server for processing.
Version 4.x ALS database server	You must set the DBCODESET environment variable to the database locale name for the ALS database server. The client application cannot perform code-set conversion.
Version 4.x ASCII database server	You must set both CLIENT_LOCALE and DB_LOCALE environment variables to the Japanese locale, ja_jp.sjis .

The remaining ALS environment variables in [Figure 17-5 on page 17-16](#) can override some or all of the basic ALS functionality. The *Informix Guide to GLS Functionality* discusses the precedence of ALS and GLS environment variables.

If you have installed a GLS database server and want your ALS database to work with this new database server, you must ensure that the GLS database server can support the ALS database. For more information, see [“Upgrading to GLS from ALS” on page 18-13](#).

Language Support in Database Servers

When a database server creates a database, it assigns a *database locale* to that database. The database locale determines the following information for the database:

- The code set whose characters are valid in any character column
- The code set whose characters are valid in the names of database objects such as databases, tables, columns, and views
- The localized order to collate data from any NCHAR and NVARCHAR columns (NLS products only)

Informix products use the database locale when they create new databases or read existing databases. A database inherits its locale from the session that creates it. When an application development tool issues the CREATE DATABASE statement, the database server creates a new database and assigns it a database locale.

The database server saves the database locale name in the system catalog of the database. Special rows in the **systables** system catalog table contain information that is used throughout the lifetime of the database for such operations as handling regular expressions, collating character strings, and ensuring proper use of code sets. At runtime, the database server maps this database locale to a locale file on the system where the database resides.

When you migrate an Informix database server whose databases contain non-English data, you need to consider whether the upgrade involves a change in Informix language support. Such a change might require any of the following changes:

- Migration of the database data
- A change in the locale that the database uses
- A change in the way locale information is stored in the **systables** system catalog table

If the locale of the data is not compatible with the language support that the database server uses, you might experience compatibility problems.

[Figure 17-8 on page 17-21](#) shows the language support that different Informix database servers use.

Figure 17-8
Language Support In Informix Database Servers

Database Server Version	Language Support
All versions	English
Dynamic Server 2000	GLS
Universal Server	GLS
Extended Parallel Server	GLS
Dynamic Server with AD and XP Options	GLS
OnLine XPS	English only
Dynamic Server, Version 7.3x	GLS
OnLine Dynamic Server SE	GLS, NLS GLS, NLS
OnLine, Version 5.0 and Version 4.1	English only
Dynamic Server, Workgroup and Developer Editions, Version 7.3x	GLS
OnLine Workgroup Server, Version 7.12 through 7.22	GLS, NLS
OnLine and SE (with ALS), Version 4.1, 5.0x, and 6.0	ALS

The following sections provide a brief summary of the support that Informix products provide for languages other than English.

Dynamic Server 2000 and Universal Server

Dynamic Server 2000 and Universal Server use the GLS feature to support languages that use both single-byte and multibyte characters. When you upgrade one of these database servers, you do *not* change language support and therefore do not need to migrate databases.

Extended Parallel Server, Dynamic Server AD/XP, and OnLine XPS

The language support that Extended Parallel Server, Dynamic Server with AD and XP Options, and OnLine XPS use depends on the version of the database server, as follows:

- Version 8.2 and later products use the GLS feature to support languages that use both single-byte and multibyte characters.
- Versions through Version 8.1x use English only.

The following table summarizes whether you need to migrate databases when you upgrade OnLine XPS or Dynamic Server with AD and XP Options.

Migrate To	Migrate From	Change Language Support?	More Information
Extended Parallel Server	Version 8.21	No	You do not need to migrate databases.
Version 8.1x	Versions through Version 8.1x	No	You do not need to migrate databases.
Version 8.21	Versions through Version 8.1x	To GLS from English only	You do not need to migrate databases.

Dynamic Server 7.3 and OnLine Dynamic Server

Dynamic Server 7.3 uses GLS. The language support that OnLine Dynamic Server uses depends on the version of the database server, as follows:

- Version 7.2x and later use the GLS feature to support languages that use both single-byte and multibyte characters.
- Version 6.0 through Version 7.1x use the NLS feature to support single-byte native languages.

- Before Version 6.0, support for languages other than English required locally customized versions of Informix products. Versions 4.x, 5.x, and 6.x of Informix ALS products provided support for multibyte characters. The standard English versions of pre-6.0 Informix products do not accommodate alternative languages. These products use the default locale of the operating system for their language support. In most cases, the default locale of the operating system is English.

The following table summarizes whether you need to migrate databases when you upgrade your Dynamic Server 7.3, OnLine Dynamic Server, or OnLine database server.

Migrate To	Migrate From	Change Language Support?	More Information
Version 7.2x and later	Version 7.2x and later	No	You do not need to migrate databases.
Version 6.0 through Version 7.1x	Version 6.0 through Version 7.1x	No	You do not need to migrate databases.
Version 7.2x and later	Version 6.0 through Version 7.1x	To GLS from NLS	“Upgrading to GLS from NLS” on page 18-6
Version 7.2x and later	Before Version 6.0 (Informix ALS products only)	To GLS from ALS	“Upgrading to GLS from ALS” on page 18-13
Version 7.2x and later	Before Version 6.0 (English-language)	No (English to English)	You do not need to migrate databases.

INFORMIX-SE

Versions of the INFORMIX-SE database server use the same language support as the corresponding OnLine Dynamic Server version. For more information, see [“Dynamic Server 7.3 and OnLine Dynamic Server” on page 17-22](#).

Dynamic Server, Workgroup and Developer Editions, and OnLine Workgroup Server

The language support that Dynamic Server, Workgroup and Developer Editions, or OnLine Workgroup Server uses depends on the version of the database server, as follows:

- Version 7.2 and later of Dynamic Server, Workgroup and Developer Editions, or OnLine Workgroup Server products use the GLS feature to support languages that use both single-byte and multibyte characters.
- Versions through Version 7.12x of OnLine Workgroup Server use the NLS feature to support single-byte native languages.

The following table summarizes whether you need to migrate databases when you upgrade Dynamic Server, Workgroup and Developer Editions, or OnLine Workgroup Server.

Migrate To	Migrate From	Change Language Support?	More Information
Version 7.2 and later	Version 7.2 and later	No	You do not need to migrate databases.
Versions through Version 7.12	Versions through Version 7.12	No	You do not need to migrate databases.
Version 7.2 and later	Versions through Version 7.12	To GLS from NLS	“Upgrading to GLS from NLS” on page 18-6

Converting to GLS

In This Chapter	18-3
Determining the Existing Locale Support	18-3
Finding the Current Database Locale	18-4
Finding Available GLS Locales on UNIX	18-4
Finding Available GLS Locales on Windows NT	18-5
Verifying GLS Database Locales	18-5
Upgrading to GLS from NLS	18-6
Determining the Locale of the NLS Database	18-6
Determining Whether a Compatible GLS Locale Exists	18-8
Opening an NLS Database	18-11
Upgrading to GLS from ALS	18-13
Migrating from Version 6.x ALS Products.	18-13
Migrating from Version 5.x ALS Products.	18-15
Migrating from Version 4.x ALS Products.	18-17
Migrating from Version 4.x ASCII Japanese Products.	18-19

In This Chapter

This chapter describes how to migrate a database that uses Native Language Support (NLS) or Asian Language Support (ALS) for its language support to the new Global Language Support (GLS) feature. For an introduction to the GLS feature, see the *Informix Guide to GLS Functionality*.

To describe how to upgrade an NLS or ALS database to run under a GLS database server, this chapter covers the following topics:

- Determining the existing locale support
- Upgrading to GLS from NLS language support
- Upgrading to GLS from ALS language support

Determining the Existing Locale Support

To verify that a GLS database server supports the database locale of your NLS or ALS database, you need the following information:

- The current database locale that your database uses
- The locales available to your GLS database server

Finding the Current Database Locale

The database locale information is stored in the system catalog of the NLS or ALS database, in special rows of the **systables** system catalog table. The exact location of this information depends on the particular database server you currently use, as follows:

- The NLS database stores database locale information in **systables** rows with **tabid** values of 90 and 91.
- The **systables** rows that an ALS database uses to store database locale depends on the version of the ALS database server. For more information, see [Figure 17-4 on page 17-15](#).

You can use a SELECT statement to query the **systables** system catalog table for the database locale information of a database.

UNIX

Finding Available GLS Locales on UNIX

To find out what GLS locales are available on your UNIX system, you can run the **glfiles** utility. This utility checks the GLS system directories and creates a file listing for the locale files that these directories contain. The file listing is in a file called **lcX.txt**, where *X* is the version of the locale files. For more information on **glfiles**, refer to the *Informix Guide to GLS Functionality*.

To run the glsfiles utility

1. Set your **INFORMIXDIR** and **PATH** environment variables to point to a GLS database server.
2. Execute the **glfiles** utility.
3. Examine the **lcX.txt** file for a GLS locale name that is compatible with your NLS locale. If you do not find an operating-system-compatible locale file for your NLS locale, try to find the closest fit.

Important: The **glfiles** utility is only available on UNIX platforms with products that support the GLS feature.



WIN NT

Finding Available GLS Locales on Windows NT

To find out which GLS locales are available on your Windows NT system, you must look in the GLS system directories. A GLS locale resides in the following file:

```
%INFORMIXDIR%\gls\lcX\lg_tr\codemodf.lco
```

In this path, **INFORMIXDIR** is the environment variable that specifies the directory in which you install the Informix product, **gls** is the subdirectory that contains the GLS system files, **X** represents the version number for the locale file format, **lg** is the 2-character language name, **tr** is the 2-character territory name that the locale supports, and **codemodf** is the condensed locale name. For more information on the location of the GLS locale files, see the *Informix Guide to GLS Functionality*.

Verifying GLS Database Locales

To find the database locales for all databases that your GLS database server supports, execute the following command in DB-Access:

```
SELECT * FROM sysmaster:sysdbslocale
```

The **sysdbslocale** system catalog table has the following two columns:

- The **dbs_dbsname** column holds the name of the database.
- The **dbs_collate** column holds the name of the database locale for the database that the **dbs_dbsname** column specifies.

For example, suppose that your GLS database server (on a UNIX system) supports the following databases: **stores_demo** and **accounting**. If the **stores_demo** database uses the default locale (**en_us.8859-1**) and the **accounting** database uses the French locale (**fr_fr.8859-1**), the preceding query might return the following output:

```
dbs_dbsnamedbs_collate
sysmasteren_US.819
sysutilsen_US.819
stores_demoen_US.819
accountingfr_FR.819
```

Upgrading to GLS from NLS

NLS databases use locales that the operating system provides and single-byte code sets. GLS databases use locales that Informix provides. Before you use an NLS database with a GLS database server, follow these steps:

1. Determine the database locale name of the current NLS database.
2. Determine which GLS locale is compatible with the current NLS database locale.
3. Open the NLS database with the GLS database server.



Important: Check the availability of the GLS locale before you open NLS databases with a GLS database server. A GLS database server cannot open an NLS database that has an unsupported locale.

This section discusses how to change to GLS language support from NLS language support when you migrate to a GLS database server from an NLS database server.



Warning: Do not overwrite your NLS database server with the GLS database server if you want to migrate a non-English NLS database to a GLS database. Migration of a non-English NLS database requires that both the NLS and GLS database servers be installed.

Determining the Locale of the NLS Database

To determine the current database locale for an NLS database, examine the site column for rows 90 and 91 of the **systables** system catalog. For more information on the use of **systables** by an NLS database, see [“NLS Locales” on page 17-11](#).

To determine the database locale of the NLS database

1. Set the appropriate NLS environment variables (see [page 17-12](#)).
2. Start an NLS database server.
3. Launch DB-Access.
4. Open your NLS database.

ODS

OVS

XPS 8.11

SE

5. Use the DB-Access Query option to execute one of the following commands, depending on your database server type.

To see the locale names of the COLLATION and CTYPE categories of a database locale for an OnLine Dynamic Server, OnLine Workgroup Server, or OnLine XPS database server, use the following SELECT statement:

```
SELECT tabname, site FROM systables
WHERE tabid = 90 OR tabid = 91
```

The value displayed in the **site** column is the name of your current NLS locale.

The following SELECT statement returns the name of the current database locale for each database that the NLS database server supports:

```
SELECT UNIQUE dbsname, collate
FROM sysmaster:systabnames
```

The **dbsname** column lists the databases on your database server. The entry in the **collate** column shows the NLS locale of the database. If the **collate** column is blank, the NLS database is an English database. ♦

To see the locale names of the COLLATION and CTYPE categories for a database locale of an SE database, use the following SELECT statement:

```
SELECT dirpath FROM systables
WHERE tabid = 90 OR tabid = 91
```

The value in the **dirpath** column is the name of your current NLS locale. ♦

If you do *not* find a matching entry in **systables**, your database uses the default locale for the operating system, which is usually U.S. English. An English NLS database does *not* require conversion. For more information, see [“To open an English NLS database” on page 18-12](#).

Determining Whether a Compatible GLS Locale Exists

To determine whether a compatible GLS locale exists for your non-English NLS database, you need to determine the following information:

- Which type of GLS locale do you want to use?
- Which GLS locales are installed on your system?

GLS products provide the following two types of locales:

- **Operating-system-independent locales.** The operating-system-independent locales are the locales that Informix has developed to provide a more portable way to support culture-specific information. With these locales, GLS products can access culture-specific information regardless of the operating system under which they run.
- **Operating-system locales.** The operating-system locales are special GLS locales that are compatible with the locales native to different operating systems. For a given language and territory, operating-system locales might have different definitions from one operating system to another. GLS products provide operating-system locales for backward compatibility with NLS databases.



***Tip:** In this manual, GLS operating-system-independent locales are usually referred to simply as “GLS locales.” The operating-system locales are also GLS locales; they share the same format as the operating-system-independent locales. However, they are useful only for NLS databases that want to continue use of their NLS locale.*

To migrate a non-English NLS database to a GLS database, you need to decide which of these types of locales to use. Your decision is not irreversible. You can also change the locale after you migrate to a GLS database server.

To use an operating-system locale

An upgrade of your NLS database to a GLS database that uses an operating-system locale requires little special action on your part.



***Important:** Use of an operating-system locale can result in incorrect results when you perform distributed queries across dissimilar environments. The locale-category definitions of one operating-system locale might differ from the locale-category definitions of the same locale on another operating system or in a GLS locale.*

To use the operating-system locale that is compatible with your current NLS database locale, perform the following steps:

1. Confirm that an operating-system locale exists for your current NLS database locale.

For more information, see [“Finding Available GLS Locales on UNIX” on page 18-4](#).

2. Set the appropriate NLS environment variables, such as **LANG** and **DBNLS**, to indicate the NLS locale.

For more information, see [“NLS Environment Variables” on page 17-12](#).

3. Open the NLS database with a GLS database server.

For more information, see [“To open a non-English NLS database” on page 18-11](#).

You do not need to perform any additional steps on the data or the system catalog. If you set the **CLIENT_LOCALE** or **DB_LOCALE** environment variables, make sure that you use the operating-system locale as the locale name.



Warning: *If your NLS database has a database locale name that is not a valid operating-system locale, or if a GLS environment variable (**CLIENT_LOCALE** or **DB_LOCALE**) is not a valid operating-system locale, the GLS database server cannot open it. You must set the **LANG** and **DBNLS** environment variables correctly for the GLS database server to open a non-English NLS database.*

To use a GLS (operating-system-independent) locale

A conversion of your non-English NLS database to a GLS database that uses a GLS (operating-system-independent) locale requires modification of the data and the system catalog. To use a GLS locale for your database, perform the following steps:

1. Determine which GLS locale is compatible with your current database locale.

For more information, see [“Finding Available GLS Locales on UNIX” on page 18-4](#).

2. Set the appropriate NLS environment variables, such as **LANG** and **DBNLS**, to indicate the NLS locale.

Do *not* set the GLS environment variables, **CLIENT_LOCALE** and **DB_LOCALE**, to the name of the new GLS locale. If you set one of these GLS environment variables, use the name of the appropriate operating-system locale. For more information on NLS environment variables, see [“NLS Environment Variables” on page 17-12](#).
3. Unload the NLS database with the **dbexport** utility.

When you unload data from an NLS database that uses an English-language locale, the resulting text files are ASCII files. When you unload data from an NLS database that uses a non-English-language locale, the resulting text files might include 8-bit characters and multibyte characters. For more information on how to use **dbexport**, see [Chapter 20, “Using the dbexport and dbimport Utilities.”](#)
4. Drop the NLS database.

You must drop the old NLS database before you import the data so that the **dbimport** utility can create a GLS database with the same name.
5. Set the GLS environment variables, **CLIENT_LOCALE** and **DB_LOCALE**, to the name of the GLS database locale that you want to use.

For more information on GLS environment variables, see [“GLS Environment Variables” on page 17-8](#).
6. Create a new GLS database and load the data into this database with the **dbimport** utility.

The **dbimport** utility determines the database locale from the **DB_LOCALE** environment variable and makes the appropriate corrections to the system catalog tables. For more information on how to use **dbimport**, see [Chapter 20, “Using the dbexport and dbimport Utilities.”](#)

You can now open the new GLS database with a GLS database server. You no longer need to set the NLS environment variables.

Opening an NLS Database

If you choose to use an operating-system locale for an NLS database, a GLS database server expects the NLS environment variables to indicate the locale of the database, as follows:

- For an English NLS database, no NLS environment variables need to be set.
The default locale on most operating systems is U.S. English.
- For a non-English NLS database, the **LANG** and **DBNLS** environment variables must be set to indicate the operating-system locale.
You can set other NLS environment variables to provide additional locale information.

With the NLS environment variables correctly set, you can open an NLS database with a GLS database server. The GLS database server automatically updates the **sysstables** system catalog table with its own database locale information when it opens the NLS database. For more information about the GLS database locale information, see [“GLS Locales” on page 17-6](#).

To open a non-English NLS database

If you choose to use an operating-system locale for your non-English NLS database, you must *always* set the NLS environment variables (such as **LANG** and **DBNLS**) appropriately. If you do *not* set these NLS environment variables, neither a GLS database server nor GLS database utilities can access the database. If you also set a GLS environment variable (**CLIENT_LOCALE** or **DB_LOCALE**), you must ensure that it is set to the name of the appropriate operating-system locale.

To open a non-English NLS database, follow these steps:

1. Set the NLS environment variables, such as **LANG** and **DBNLS**, to indicate use of an operating-system locale for the database.
For more information on NLS environment variables, see [Figure 17-3 on page 17-12](#).
2. You can optionally set the GLS environment variables, **CLIENT_LOCALE** and **DB_LOCALE**, to the name of the operating-system locale for the database.
For more information on GLS environment variables, see [Figure 17-1 on page 17-8](#).

3. When the GLS database server opens an NLS database, the database server automatically updates the **systables** system catalog table, as follows:

- In the row whose **tabid** value is 90, the database server changes the **tabname** value from NLSCOLL to GL_COLLATE.
- In the row whose **tabid** value is 91, the database server changes the **tabname** value from NLSCTYPE to GL_CTYPE.
- The database server creates a condensed database locale name from the locale name in rows 90 and 91 of **systables**.

If a locale exists that corresponds to the condensed name, the database server stores the condensed locale name in rows 90 and 91. If the existing locale does *not* have a condensed locale name, the database server does not change the locale name in rows 90 and 91.

4. If the GLS locale defines a localized collation order, the GLS database server upgrades the data type of all CHAR and VARCHAR columns in the system catalog tables to NCHAR and NVARCHAR.

If the GLS locale does not define a localized collation order, the database server does not upgrade the data type of all CHAR and VARCHAR columns in the system catalog tables.



Warning: If the **systables** rows with **tabid** values of 90 and 91 contain an invalid operating-system locale name, the GLS database server cannot open the NLS database.

To open an English NLS database

GLS products use the default locale, U.S. English, automatically. Most NLS products also use U.S. English by default because the default locale on most operating systems is U.S. English. Therefore, you are not required to set the GLS environment variables, **CLIENT_LOCALE** and **DB_LOCALE**, before you open an English NLS database with a GLS database server. (If you do set these environment variables, they must be set to the name of the default GLS locale.)

When you open an English NLS database, the GLS database server changes the **systables** system catalog table, as follows:

- Creates a row with the **tabname** of GL_COLLATE and the **tabid** of 90.
- Creates a row with the **tabname** of GL_CTYPE and the **tabid** of 91.
- Stores a condensed locale name for the default locale in the **site** column of rows 90 and 91.

The default locale, U.S. English, does not define a localized collation order. Therefore, the GLS database server does *not* upgrade CHAR and VARCHAR columns in the system catalog tables to NCHAR and NVARCHAR. For English databases, the database server uses code-set collation order for collation of data in *all* character columns (in the system catalog tables or user-defined tables).

Upgrading to GLS from ALS

The Informix ALS products include ALS locales, which contain culture-specific information and the multibyte code sets that Asian languages require. When a GLS database server opens an ALS database, the database server automatically upgrades the database locale name to a GLS locale name from its ALS locale name and then copies this GLS locale name into rows 90 and 91 of the **systables** system catalog table. Therefore, you do not need to change the database locale name to a GLS locale from its ALS locale when you upgrade to a GLS database from an ALS database.

This section discusses how to upgrade to GLS language support from ALS language support for the following versions of ALS databases:

- Version 6.x ALS databases
- Version 5.x ALS databases
- Version 4.x ALS databases
- Version 4.x ASCII Japanese databases

Migrating from Version 6.x ALS Products

The GLS database server automatically migrates a Version 6.x ALS database when it opens the database. Informix Version 6.x ALS databases store locale information in the **systables** system catalog table in the rows whose **tabid** values are 95 and 96. When you use a GLS database server to open a Version 6.x ALS database, the database server copies this locale information to rows in **systables** whose **tabid** values are 90 and 91, respectively.

To open a Version 6.x ALS database

1. Verify that the current ALS locale of your database is available to your GLS database server.

[Figure 18-1](#) shows the Version 6.x ALS locales. For more information on how to determine which GLS locales are installed with your database server, see [“Finding Available GLS Locales on UNIX” on page 18-4](#). If a compatible GLS locale is *not* available, you must obtain this GLS locale *before* you proceed with the migration.
2. Set the GLS environment variables, **CLIENT_LOCALE** and **DB_LOCALE**, to the new GLS locale for the ALS database.

For more information on GLS environment variables, see [Figure 17-1 on page 17-8](#).
3. Set the **GLS8BITFSYS** environment variable to the same value as the **ALS8BITFSYS** environment variable.
4. When the GLS database server opens the Version 6.x ALS database, the database server automatically updates the **systables** system catalog table, as follows:
 - In the row whose **tabid** value is 90, the database server copies the data from the row whose **tabid** is 95 and assigns the **tabname** value of GL_COLLATE to row 90.
 - In the row whose **tabid** value is 91, the database server copies the data from the row whose **tabid** is 96 and assigns the **tabname** value of GL_CTYPE to row 91.
 - The database server creates a condensed database locale name from the locale name in rows 90 and 91 of **systables**.

If a locale exists that corresponds to the condensed name, the database server stores the condensed locale name in rows 90 and 91. If the existing locale does *not* have a condensed locale name, the database server does not change the locale name in rows 90 and 91.

Figure 18-1
Version 6.x ALS
Locale Names

Asian Language	6.x ALS Locale Name
English	en_us.8859-1
China	zh_CN.gb
Korea	ko_KR.ksc
Japan	ja_JP.sjis ja_JP.sjis-s ja_JP.ujis
Taiwan	zh_TW.big5 zh_TW.sbig5 zh_TW.ccdc zh_TW.cccii

Migrating from Version 5.x ALS Products

The GLS database server automatically migrates a Version 5.x ALS database when it opens the database. Version 5.x ALS databases store code-set aliases in the database.



Warning: If your 5.x ALS database uses a **DBCODASET** of **sjis** and has user-defined double-byte characters that are mapped between 0xf040 and 0xfcfc, the following restrictions apply:

- Do not open the 5.x ALS database with a GLS database server.
Instead, use the **dbexport** and **dbimport** utilities or the **UNLOAD** and **LOAD** commands in DB-Access to migrate the 5.x ALS database to a GLS database.
- Do not issue the SQL statement **CREATE DATABASE** from 5.x ALS client applications that use a **DBCODASET** of **sjis**.
Instead, use a GLS version of DB-Access to create new databases. However, you can use 5.x ALS client applications to create new tables, indexes, views, and so forth.

To open a Version 5.x ALS database

1. Verify that the current ALS code-set alias of your database is available to your GLS database server.

[Figure 18-2](#) shows the GLS locale names for the Version 5.x code-set aliases. For more information on how to determine which GLS locales are installed for your database server, see [“Finding Available GLS Locales on UNIX” on page 18-4](#). If the compatible GLS locale is *not* available, you must obtain this GLS locale *before* you proceed with the migration.

2. Set the GLS environment variables **CLIENT_LOCALE** and **DB_LOCALE** to the new GLS locale for the ALS database.

For more information on GLS environment variables, see [Figure 17-1 on page 17-8](#).

3. When the GLS database server opens the Version 5.x ALS database, the database server automatically performs the following steps:

- Maps the 5.x code-set alias to its GLS locale name, as [Figure 18-2](#) shows.

If the code-set alias stored in the 5.x ALS database is *not* one of these aliases, the conversion fails. The conversion also fails if the code-set alias was customized.

- Stores the mapped GLS locale name to rows in **systables** whose **tabid** values are 90 and 91.

For more information on how a GLS database stores information in **systables**, see [“GLS Locales” on page 17-6](#).

Figure 18-2
Locale-Name Conversion for a Version 5.x ALS Database

Version 5.x ALS Code-Set Name	GLS Locale Name
big5	zh_TW.big5
sbig5	zh_TW.sbig5
ccdc	zh_TW.ccdc
cccii	zh_TW.cccii
8859-1	en_us.8859-1
gb2312	zh_CN.gb

Migrating from Version 4.x ALS Products

The GLS database server automatically migrates a Version 4.x ALS database when it opens the database. Informix Version 4.x ALS databases store locale information in the **systables** system catalog table in a row whose **tabid** value is 98. When you use a GLS database server to open a Version 4.x ALS database, the database server copies this locale information to rows in **systables** whose **tabid** values are 90 and 91.



Warning: If your 4.x ALS database uses a **DBCODESET** of **sjis** and has user-defined double-byte characters that are mapped between 0xf040 and 0xfcfc, the following restrictions apply:

- **Do not open this 4.x ALS database with a GLS database server.**
*Instead, use the **dbexport** and **dbimport** utilities or UNLOAD and LOAD commands in DB-Access to migrate the 4.x ALS database to a GLS database.*
- **Do not issue the SQL statement CREATE DATABASE from 4.x ALS client applications that use a **DBCODESET** of **sjis**.**
Instead, use a GLS version of DB-Access to create new databases. However, you can use 4.x ALS client applications to create new tables, indexes, views, and so forth.

To open a Version 4.x ALS database

1. Verify that the current ALS code-set name of your database is available to your GLS database server.
[Figure 18-3](#) shows the GLS locale names for the Version 4.x code-set names. For more information on how to determine which GLS locales are installed for your database server, see [“Finding Available GLS Locales on UNIX” on page 18-4](#). If the compatible GLS locale is *not* available, you must obtain this GLS locale *before* you proceed with the migration.
2. Set the GLS environment variables **CLIENT_LOCALE** and **DB_LOCALE** to the new GLS locale for the ALS database.
For more information on GLS environment variables, see [Figure 17-1 on page 17-8](#).
3. When the GLS database server opens the Version 4.x ALS database, the database server automatically performs the following steps:
 - Maps the 4.x code-set name that is stored in row 98 of the **systables** system catalog table to a GLS locale name, as [Figure 18-3](#) shows.
If the code set that is stored in the 4.x ALS database is *not* one of these names, the conversion fails.
 - Stores the mapped GLS locale name in the **site** column of the **systables** rows whose **tabid** values are 90 and 91.
For more information on how a GLS database stores information in **systables**, see [“GLS Locales” on page 17-6](#).

Figure 18-3
*Locale-Name
 Conversion for a
 Version 4.x ALS
 Database*

Version 5.x ALS Code-Set Name	GLS Locale Name
ascii	en_us.8859-1
gb	zh_CN.gb
ksc	ko_KR.ksc
ksc1	ko_KR.ksc
big5	zh_TW.big5
sbig5	zh_TW.sbig5
ccdc	zh_TW.ccdc
cccii	zh_TW.cccii

Migrating from Version 4.x ASCII Japanese Products

The GLS database server cannot automatically upgrade 4.x ASCII databases (Japanese-language version of 4.x ALS) to GLS databases. You must use the **dbexport** and **dbimport** utilities or the UNLOAD and LOAD commands to unload the 4.x ASCII database and then load it into a GLS database server.



Warning: Do not issue the SQL statement `CREATE DATABASE` from 4.x ASCII client applications. Instead, use a GLS version of DB-Access to create new databases. However, you can use 4.x ASCII client applications to create new tables, indexes, views, and so forth.

Reverting from GLS

In This Chapter	19-3
Using the onmode Utility for Reversion.	19-3
Reverting to NLS from GLS	19-3
Reverting to an NLS Database Server	19-4
Reverting to an NLS Version of the SE Database Server	19-5
Reverting to ALS from GLS	19-5
Reverting to Version 6.x ALS	19-6
Reverting to Version 5.x ALS	19-7
Reverting to Version 4.x ALS or ASCII Products	19-8



In This Chapter

This chapter describes how to revert to a database that uses Native Language Support (NLS) or Asian Language Support (ALS) for its language support from the Global Language Support (GLS) feature.

Using the `onmode` Utility for Reversion

The **onmode** utility with its **-b** option reverts a GLS database with the following syntax:

```
onmode -b reversion_level
```

In the preceding syntax, *reversion_level* indicates to which version of the database server (and hence to which language support) **onmode** reverts all databases that the GLS database server supports. For more information on the **onmode** utility, see your GLS database server *Administrator's Guide*.

The following sections describe how to use **onmode** to revert to an NLS database or an ALS database from a GLS database.

Reverting to NLS from GLS

During the reversion process, the **onmode** utility moves the database locale information in **systables** to its NLS location from its GLS location.



Warning: Before you revert a GLS database to an NLS database, verify that the GLS databases use operating-system locales. Also verify that your operating system supports these locales. An NLS database server can open a reverted GLS database only if the operating system supports the database locale.

This section discusses how to revert a GLS database to an NLS database for use with the following NLS database servers:

- OnLine Dynamic Server
- OnLine Workgroup Server
- SE

ODS

OVS

Reverting to an NLS Database Server

You can use the **onmode** utility to revert to an NLS version of OnLine Dynamic Server or OnLine Workgroup Server. OnLine Dynamic Server 6.0, 7.1, 7.1UD1, and 7.10.UCx and OnLine Workgroup Server 7.12 support only the NLS feature for language support.

To revert to an NLS database server from a GLS database server

1. Use the **onmode** utility with its **-b** option, as follows:

```
onmode -b version_number
```

In the previous syntax, *version_number* is one of the values that the following table shows.

OnLine Target Version	<i>version_number</i> Value
6.0	6.0
7.1UC1	7.1
7.1UD1, 7.11, 7.12, 7.13, or 7.14	7.1UD1

2. Set the appropriate NLS environment variables.

An NLS database server ignores the **CLIENT_LOCALE**, **DB_LOCALE**, and **SERVER_LOCALE** environment variables. For more information on NLS environment variables, see [Figure 17-3 on page 17-12](#).

If the GLS database uses the default locale (U.S. English) and none of its user-defined tables have NCHAR or NVARCHAR columns, the **onmode** utility performs the following actions to revert the database:

- It deletes the locale entries in **systables**.
- It upgrades any NCHAR and NVARCHAR columns in the system catalog tables to CHAR and VARCHAR columns, respectively.
- It remakes the indexes on the system catalog tables.

If the GLS database is a non-English database (it uses any nondefault locale), or if any user-defined table has NCHAR or NVARCHAR columns, the **onmode** utility changes the locale entries in **systables** to the NLS format (**tablename** from rows 90 and 91 changes from GL_COLLATE and GL_CTYPE to NLSCOLL and NLSCTYPE, respectively). You do not need to remake the indexes.

SE

Reverting to an NLS Version of the SE Database Server

SE does not provide a direct path for reverting to earlier NLS versions from a GLS version of the product. For more information, refer to [Chapter 13, “Migrating Between INFORMIX-SE Database Servers and Converting C-ISAM Files.”](#)

Reverting to ALS from GLS

During the reversion process, the **onmode** utility moves the database locale information in the **systables** system catalog table to its ALS location from its GLS location.



Warning: Before you revert a GLS database to an ALS database, verify that the GLS database uses a locale that the ALS database server supports. An ALS database server can open a reverted GLS database only if the database locale is one that the database server supports.

This section discusses how to revert a GLS database to an ALS database for use with the following Informix ALS products:

- Version 6.x ALS database servers
- Version 5.x ALS database servers
- Version 4.x ALS database servers

Reverting to Version 6.x ALS

Use the **-b 6.0A** option of the **onmode** utility to revert a GLS database to a Version 6.x ALS database. The **onmode** utility performs the following actions to revert the database:

- It moves the locale entries in **systables** to the 6.0 ALS database location (from rows 90 and 91 to rows 95 and 96).
- It upgrades NCHAR columns in the system catalog table to CHAR columns.
- It remakes the indexes in the system catalog tables.

Important: Only NCHAR and NVCHAR columns are affected by a collation order. A collation order, specified in a locale, is ignored if the table does not contain NCHAR or NVCHAR.



To revert to a Version 6.x ALS database from a GLS database

1. Verify that the Version 6.x ALS database server supports the current GLS database locale.
[Figure 17-2 on page 17-9](#) shows the Version 6.x ALS locales. If the GLS locale is *not* one of these locales, the reversion fails.
2. Choose one of the following methods to handle any NCHAR or NVARCHAR columns from your database:
 - Replace NCHAR columns with CHAR columns and NVARCHAR columns with VARCHAR columns.
This solution means that you lose the locale-specific data, but it ensures that the 6.x ALS database server supports the remaining data.
 - Drop the NCHAR or NVARCHAR columns from your database.
This solution means that the database server no longer collates character data in a localized order.

3. Use the following **onmode** command to perform the reversion:

```
onmode -b 6.0A
```

4. Set the appropriate ALS environment variables.

For more information, see [“ALS Environment Variables” on page 17-16](#).



Warning: If the database has *NCHAR* or *NVARCHAR* in any user-defined table, **onmode** fails.

Suppose you upgrade a 6.x ALS database to a GLS database and then revert it back to 6.x ALS. This type of reversion does not pose a problem because the database server did not remove the original ALS locale entries in rows 95 and 96 when it upgraded the ALS database to a GLS database.

However, when you revert a database that a GLS database server creates, the **onmode** utility might encounter some problems. A GLS database server and a Version 6.x ALS database server can support the same Asian locales. The **onmode** utility moves the locale information from rows 90 and 91 to rows 95 and 96 of **systables**. However, the GLS database stores its database locale name in a condensed format, which a Version 6.x ALS database server does not recognize.

For example, both Version 6.x ALS products and GLS products support the default locale, U.S. English. However, Version 6.x ALS products store the name of this locale as **en_us.8859-1**, while GLS products (on a UNIX platform) store it as **en_us.819**, which 6.x ALS products do not understand. You might need to update the name of the database locale in **systables** to a form that the Version 6.x ALS database server supports.

Reverting to Version 5.x ALS

Use the **-b 5.0** option of the **onmode** utility to revert a GLS database to a Version 5.x ALS database.



Important: You cannot revert to a 5.x ALS database server from a GLS database server automatically.

To revert to a 5.x ALS database from a GLS database

1. Verify that the Version 5.x ALS database server supports the current GLS database locale.
[Figure 17-2 on page 17-9](#) shows the valid GLS locales for a Version 5.x ALS database server. If the GLS locale is *not* one of these locales, the reversion fails.
2. Verify that the current GLS locale is supported in Version 5.x ALS. (See [Figure 17-2 on page 17-9](#).)
3. Choose one of the following methods to handle any NCHAR or NVARCHAR columns from your database:
 - Drop all NCHAR and NVARCHAR columns from the database.
This solution means that you lose the locale-specific data, but it ensures that the 5.x ALS database server supports the remaining data.
 - Upgrade the NCHAR columns to CHAR and NVARCHAR columns to VARCHAR.
This solution means that the database server no longer collates character data in a localized order.
4. Use the following **onmode** command to revert the GLS databases:

```
onmode -b 5.0
```
5. Set the appropriate ALS environment variables.
For more information, see [Figure 17-5 on page 17-16](#).

All databases that use OnLine 5.0 have to be in **en_us.8859-1** or **en_us.819** for both CTYPE and COLLATE.

If your GLS database contains locale-specific data (NCHAR and NVARCHAR columns), and you revert to a database server that does not support non-English data, you lose access to this data.



Warning: The **onmode** utility fails when it attempts to revert a GLS database that contains NCHAR or NVARCHAR columns to a Version 5.x ALS database.

Reverting to Version 4.x ALS or ASCII Products

GLS products do not support reversion to Version 4.x ALS products or to Version 4.x ASCII products.

Data Migration Utilities

- Chapter 20 Using the dbexport and dbimport Utilities
- Chapter 21 Using the dbload Utility
- Chapter 22 Using the dbschema Utility
- Chapter 23 Using the LOAD and UNLOAD Statements
- Chapter 24 Using the onmode Utility
- Chapter 25 Using the onunload and onload Utilities
- Chapter 26 Using the onxfer Utility



Using the dbexport and dbimport Utilities

In This Chapter	20-3
Syntax of the dbexport Command	20-5
Syntax of the dbimport Command	20-11
Simple Large Objects	20-19
Database Locale Changes	20-20

In This Chapter

This chapter describes the **dbexport** and **dbimport** utilities and how to use them. You can use **dbexport** and **dbimport** with the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE
- OnLine

The **dbexport** utility unloads a database into text files for later import into another database and creates a schema file. The **dbimport** utility creates and populates a database from text files. You can use the schema file with **dbimport** to re-create the database schema in another Informix environment. You can edit the schema file to modify the database that **dbimport** creates. The **dbexport** and **dbimport** utilities support Dynamic Server 2000 and Universal Server data types.

Informix stores dates in four-digit year dates. By default, **dbexport** exports dates in four-digit year dates unless the environment variable **DBDATE** is set to “mdy2”. Informix does not recommended this setting for exporting a database because data imported back into the database depends on either the **DBCENTURY** environment variable, if set, or the current century if **DBCENTURY** is not set.

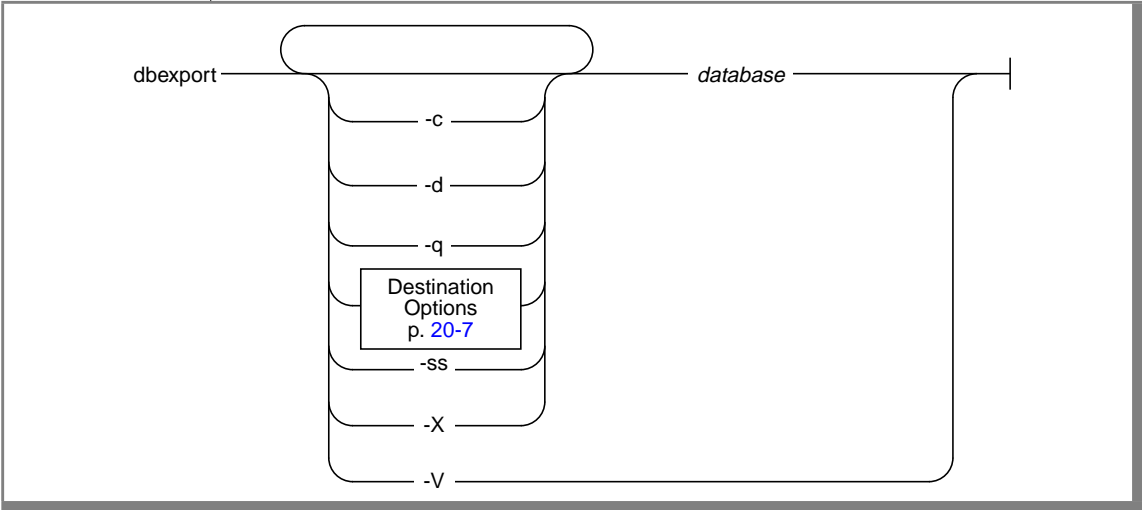


The **dbexport** and **dbimport** utilities are not part of the following database servers:

- Dynamic Server with AD and XP Options
- Extended Parallel Server
- OnLine XPS

***Tip:** The examples in this chapter apply to all other Informix database servers unless they are marked specifically with an icon for SE.*

Syntax of the dbexport Command



Element	Purpose	Key Considerations
-c	Makes dbexport complete exporting unless a fatal error occurs.	References: For details on this option, see “Errors” on page 20-7 .
-d	Makes dbexport export simple-large-object descriptors only, not simple-large-object data.	References: For more information about simple-large-object descriptors, refer to the <i>Guide to the Optical Subsystem</i> . Restrictions: Not supported by SE.
-q	Suppresses the display of error messages, warnings, and generated SQL data-definition statements.	None.
-ss	Generates database server-specific information for all tables in the specified database.	References: For details on this option, see “Server-Specific Information” on page 20-7 .

(1 of 2)

Element	Purpose	Key Considerations
-X	Recognizes HEX binary data in character fields.	None.
-V	Displays product version information.	None.
database	Specifies the name of the database that you want to export.	<p>Additional Information: If your locale is set to use multibyte characters, you can use multibyte characters for the database name.</p> <p>References: If you want to use more than the simple name of the database, refer to the Database Name section of the <i>Informix Guide to SQL: Syntax</i>.</p>

(2 of 2)

GLS

You must have DBA privileges or log in as user **informix** to export a database.

When the environment variables are set correctly, as described in the *Informix Guide to GLS Functionality*, **dbexport** can handle foreign characters in data and export the data from GLS databases. For more information, refer to “[Database Renaming](#)” on page 20-17. ♦

You can use delimited identifiers with the **dbexport** utility. The utility detects database objects that are keywords, mixed case, or have special characters and places double quotes around them.

In addition to the data files and the schema file, **dbexport** creates a file of messages called **dbexport.out** in the current directory. This file contains error messages, warnings, and a display of the SQL data definition statements that it generates. The same material is also written to the standard output unless you specify the **-q** option.

During the export, the database is locked in exclusive mode. If **dbexport** cannot obtain an exclusive lock, it displays a diagnostic message and exits.

Termination of dbexport

You can press the INTERRUPT key at any time to cancel **dbexport**. The **dbexport** utility asks for confirmation before it terminates.

Errors

The **-c** option tells **dbexport** to complete exporting unless a fatal error occurs. Even if you use the **-c** option, **dbimport** interrupts processing if one of the following fatal errors occurs:

- Unable to open the tape device specified
- Bad writes to the tape or disk
- Invalid command parameters
- Cannot open database or no system permission

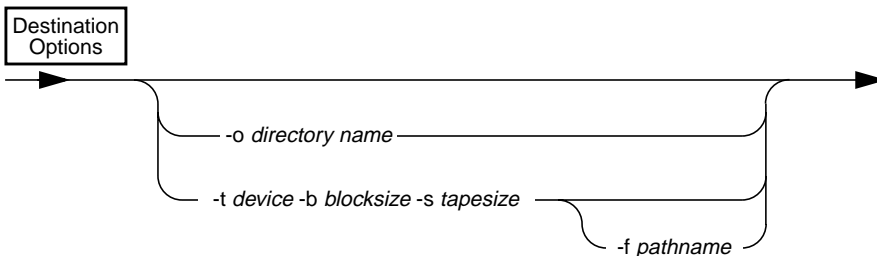
Server-Specific Information

The **-ss** option generates server-specific information. The **-ss** option specifies initial- and next-extent sizes, fragmentation information if the table is fragmented, the locking mode, the dbspace for a table, the blobspace for any simple large objects, and the dbspace for any smart large objects.

SE

For SE databases, the **-ss** option generates the pathname of each table that is in a path other than the database directory. ♦

Destination Options



Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies, in kilobytes, the block size of the tape device.	None.
-f <i>pathname</i>	Specifies the pathname where you want the schema file stored, if you are storing the data files on tape.	Additional Information: The pathname can be a complete pathname or simply a filename. If only a filename is given, the file is stored in the current directory.
-o <i>directory name</i>	Names the directory on disk in which dbexport creates the <i>database.exp</i> directory. This directory holds the data files and the schema file that dbexport creates for the <i>database</i> .	Restrictions: The directory specified as <i>directory name</i> must already exist.
-s <i>tapesize</i>	Specifies, in kilobytes, the amount of data that you can store on the tape.	Restrictions: The tape size is limited to 2,097,151 kilobytes. The limit is required because of the way dbexport and dbimport track their positions into the tape.
-t <i>device</i>	Specifies the pathname of the tape device where you want the text files and, possibly, the schema file stored.	Restrictions: The -t option does not allow you to specify a remote tape device.

When you write to disk, **dbexport** creates a subdirectory, *database.exp*, in the directory that the **-o** option specifies. The **dbexport** utility creates a file with the *.unl* extension for each table in the database. The schema file is written to the file *database.sql*. The *.unl* and *.sql* files are stored in the *database.exp* directory.

If you do not specify a destination for the data and schema files, the subdirectory *database.exp* is placed in the current working directory.

When you write the data files to tape, you can use the **-f** option to store the schema file to disk. You are not required to name the schema file *database.sql*. You can give it any name.

SE

The following **dbexport** command creates a **reports.exp** subdirectory in the current directory. It then unloads the **reports** database in the **turku** directory on the SE database server called **finland** and places the resulting files in the **reports.exp** directory, as follows:

```
dbexport //finland/turku/reports
```



UNIX

For other non-SE database servers, the same command is as follows:

```
dbexport //finland/reports
```

The following command exports the database **stores_demo** to tape with a block size of 16 kilobytes and a tape capacity of 24,000 kilobytes. The schema file is written to **/tmp/stores_demo.imp**.

```
dbexport -t /dev/rmt0 -b 16 -s 24000 -f /tmp/stores_demo.imp stores_demo
```

The following command exports the same **stores_demo** database to the directory named **/work/exports/stores_demo.exp**. The resulting schema file is **/work/exports/stores_demo.exp/stores_demo.sql**.

```
dbexport -o /work/exports stores_demo
```



WIN NT

For Windows NT, the following command exports the database **stores_demo** to tape with a block size of 16 kilobytes and a tape capacity of 24,000 kilobytes. The schema file is written to **C:\temp\stores_demo.imp**.

```
dbexport -t \\.\TAPE2 -b 16 -s 24000 -f  
C:\temp\stores_demo.imp stores_demo
```

The following command exports the same **stores_demo** database to the directory named **D:\work\exports\stores_demo.exp**. The resulting schema file is **D:\work\exports\stores_demo.exp\stores_demo.sql**.

```
dbexport -o D:\work\exports stores_demo
```



IDS 2000

IUS

Contents of the Schema File

The schema file contains the SQL statements that you need to re-create the exported database. You can edit the schema file to modify the schema of the database.

The schema file supports all Dynamic Server 2000 and Universal Server data types. ♦

If you use the **-ss** option, the schema file contains server-specific information, such as initial- and next-extent sizes, fragmentation information, lock mode, the dbspace where each table resides, the blob space where each simple-large-object column resides, and the dbspace for smart large objects. The following information is not retained:

- Logging mode of the database (For information about logging modes, refer to the *Informix Guide to SQL: Reference*.)
- The starting values of SERIAL columns

The statements in the schema file that create tables, views, indexes, roles, and grant privileges do so with the name of the user who originally created the database. In this way, the original owner retains DBA privileges for the database and is the owner of all the tables, indexes, and views. In addition, the person who executes the **dbimport** command also has DBA privileges for the database.

The schema file that **dbexport** creates contains comments, enclosed in curly braces, with information about the number of rows, columns, and indexes in tables, and information about the unload files. The **dbimport** utility uses the information in these comments to load the database.

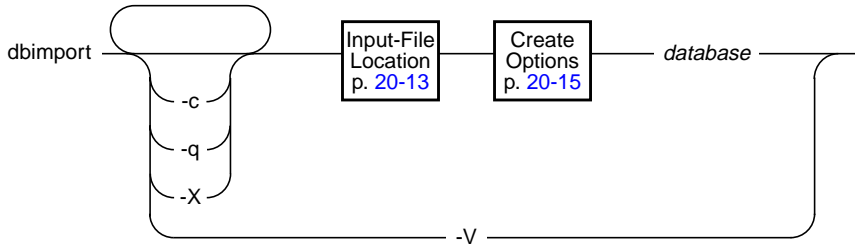
Warning: Do not delete any comments in the schema file. Informix strongly recommends that you do not change any existing comments or add any new comments, or the **dbimport** might abort or produce unpredictable results.

The number of rows should match in the unload file and the corresponding unload comment in the schema file. If you change the number of rows in the unload file but not the number of rows in the schema file, a mismatch occurs.

Tip: If you delete some rows from an unload file, update the comment in the schema file with the correct number of rows for that unload file. Then **dbimport** will be successful.



Syntax of the dbimport Command



Element	Purpose	Key Considerations
-c	Instructs dbimport to complete importing even though it encounters certain nonfatal errors.	References: For more information, refer to “Errors and Warnings” on page 20-12 .
-q	Suppresses the display of error messages, warnings, and generated SQL data-definition statements.	None.
-V	Displays product version information.	None.
-X	Recognizes HEX binary data in character fields.	None.
database	Specifies the name of the database to create.	Additional Information: If you want to use more than the simple name of the database, refer to the Database Name section of the <i>Informix Guide to SQL: Syntax</i> .

The **dbimport** utility can use files from the following location options:

- All input files are located on disk.
- All input files are located on tape.
- The schema file is located on disk, and the data files are located on tape.



Important: Do not put comments into your input file. Comments might cause unpredictable results when the **dbimport** utility reads them.

SE

The **dbimport** utility supports the following options for a new Informix database server (excluding SE):

- Create an ANSI-compliant database (includes unbuffered logging).
- Establish transaction logging for a database (unbuffered or buffered logging).
- Specify the dbspace where the database will reside.

The **dbimport** utility supports the following options for a new SE database:

- Create an ANSI-compliant database (ANSI-compliant logging).
- Establish transaction logging for a database (unbuffered logging). ♦

The user who runs **dbimport** is granted the DBA privilege on the newly created database. The **dbimport** process locks each table as it is being loaded and unlocks the table when the loading is completed.

GLS

When the GLS environment variables are set correctly, as the *Informix Guide to GLS Functionality* describes, **dbimport** can import data into database versions that support GLS. ♦

Termination of dbimport

To cancel **dbimport**, press the INTERRUPT key at any time. The **dbimport** program asks for confirmation before it terminates.

Errors and Warnings

If you include the **-c** option, **dbimport** ignores the following errors:

- A data row that contains too many columns
- Inability to put a lock on a table
- Inability to release a lock

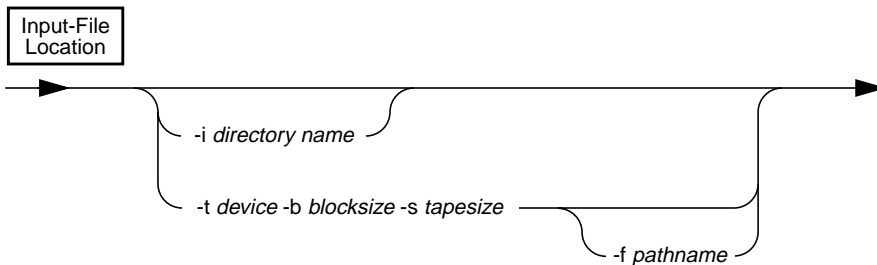
Even if you use the **-c** option, **dbimport** interrupts processing if one of the following fatal errors occurs:

- Unable to open the tape device specified
- Bad writes to the tape or disk
- Invalid command parameters
- Cannot open database or no system permission
- Cannot convert the data

The **dbimport** utility creates a file of messages called **dbimport.out** in the current directory. This file contains any error messages and warnings that are related to **dbimport** processing. The same information is also written to the standard output unless you specify the **-q** option.

Input-File Location Options

The input-file location tells **dbimport** where to look for the *database.exp* directory, which contains the files that **dbimport** will import. If you do not specify an input-file location, **dbimport** looks for data files in the directory *database.exp* under the current directory and for the schema file in *database.exp/database.sql*.



Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies, in kilobytes, the block size of the tape device.	Restrictions: If you are importing from tape, you must use the same block size that you used to export the database.
-f <i>pathname</i>	Specifies where dbimport can find the schema file to use as input to create the database when the data files are read from tape.	Additional Information: If you use the -f option to export a database, you typically use the same pathname that you specified in the dbexport command. If you specify only a filename, dbimport looks for the file in the .exp subdirectory of your current directory.
-i <i>directory name</i>	Specifies the complete pathname on disk of the database.exp directory, which holds the input data files and schema file that dbimport uses to create and load the new database. The directory name should be the same as the database name.	Additional Information: This directory should be the same directory that you specified with the dbexport -o option. If you change the directory name, you also rename your database.
-s <i>tapesize</i>	Specifies, in kilobytes, the amount of data that you can store on the tape.	Restrictions: If you are importing from tape, you must use the same tape size that you used to export the database.
-t <i>device</i>	Specifies the pathname of the tape device that holds the input files.	Restrictions: The -t option does <i>not</i> allow you to specify a remote tape device.

UNIX

The following command imports the **stores_demo** database from a tape with a block size of 16 kilobytes and a capacity of 24,000 kilobytes. The schema file is read from **/tmp/stores_demo.imp**.

```
dbimport -c -t /dev/rmt0 -b 16 -s 24000 -f
/tmp/stores_demo.imp stores_demo
```

The following command imports the **stores_demo** database from the **stores_demo.exp** directory under the **/work/exports** directory. The schema file is assumed to be **/work/exports/stores_demo.exp/stores_demo.sql**.

```
dbimport -c -i /work/exports stores_demo
```



WIN NT

The following command imports the **stores_demo** database from a tape with a block size of 16 kilobytes and a capacity of 24,000 kilobytes. The schema file is read from **C:\temp\stores_demo.imp**.

```
dbimport -c -t \\.\TAPEDRIVE -b 16 -s 24000 -f
C:\temp\stores_demo.imp stores_demo
```

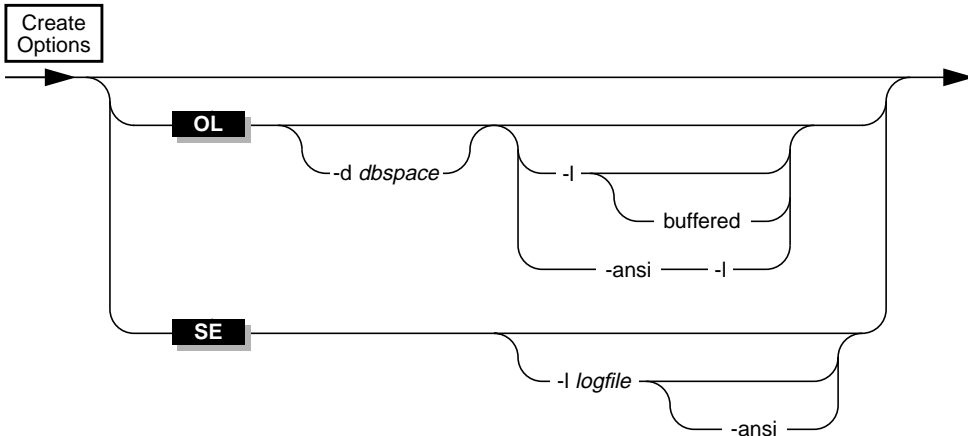
The following command imports the **stores_demo** database from the **stores_demo.exp** directory under the **D:\work\exports** directory. The schema file is assumed to be

D:\work\exports\stores_demo.exp\stores_demo.sql.

```
dbimport -c -i D:\work\exports stores_demo
```

♦

Create Options



Element	Purpose	Key Considerations
-ansi	Creates an ANSI-compliant database in which the ANSI rules for transaction logging are enabled.	Additional Information: If you specify the -ansi option, you must also specify the -l logfile option. For more information about ANSI-compliant databases, refer to the <i>Informix Guide to SQL: Reference</i> .
-d dbspace	Names the dbspace where the database is created. The default dbspace location is the rootdbs.	Additional Information: For SE, the database is always in the current directory.
-l	Establishes unbuffered transaction logging for the imported database.	References: For more information, refer to “Database-Logging Mode” on page 20-17.
-l buffered	Establishes buffered transaction logging for the imported database.	References: For more information, refer to “Database-Logging Mode” on page 20-17.
-l logfile	Establishes transaction logging for the imported database and specifies the name of the transaction-log file.	Restrictions: For SE, the <i>logfile</i> filename must be an absolute pathname or in the current directory. References: For more information, refer to “Database-Logging Mode” on page 20-17.

UNIX

The following command imports the **stores_demo** database from the **/usr/informix/port/stores_demo.exp** directory. The new database is ANSI compliant, and the transaction-log file is specified as **stores_demo.log** in **/usr/work**.

```
dbimport -c stores_demo -i /usr/informix/port -l
/usr/work/stores_demo.log -ansi
```



WIN NT

The following command imports the **stores_demo** database from the **C:\USER\informix\port\stores_demo.exp** directory. The new database is ANSI compliant, and the transaction-log file is specified as **stores_demo.log** in **C:\USER\work**.

```
dbimport -c stores_demo -i C:\USER\informix\port -l
C:\USER\work\stores_demo.log -ansi
```



Database-Logging Mode

The logging mode is not retained in the schema file. You can specify any of the following options when you use **dbimport** to import a database:

- ANSI-compliant database with unbuffered logging
- Unbuffered logging
- Buffered logging

For more information, refer to [“Create Options” on page 20-15](#).

The **-l** options are equivalent to the logging clauses of the CREATE DATABASE statement, as follows:

- The **-l** option is equivalent to the WITH LOG clause.
- The **-l buffered** option is equivalent to the WITH BUFFERED LOG.
- The **-l logfile** option is equivalent to the WITH LOG IN clause. ♦

For more information about the CREATE DATABASE statement, see the *Informix Guide to SQL: Syntax*.

Database Renaming

The **dbimport** utility gives the new database the same name as the database that you exported. If you export a database to tape, you cannot change its name when you import it with **dbimport**.

If you export a database to disk, you can change the database name.

UNIX

To change the database name to newname on UNIX

In the following example, assume that **dbexport** unloaded the database **stores_demo** into the directory **/work/exports/stores_demo.exp**. Thus, the data files (the **.unl** files) are stored in **/work/exports/stores_demo.exp**, and the schema file is **/work/exports/stores_demo.exp/stores_demo.sql**.

1. Change the name of the **.exp** directory. That is, change **/work/exports/stores_demo.exp** to **/work/exports/newname.exp**.
2. Change the name of the schema file. That is, change **/work/exports/stores_demo.exp/stores_demo.sql** to **/work/exports/stores_demo.exp/newname.sql**. Do not change the names of the **.unl** files.
3. Import the database with the following command:

```
dbimport -i /work/exports newname
```



WIN NT

To change the database name to newname on Windows NT

In the following example, assume that **dbexport** unloaded the database **stores_demo** into the directory **D:\work\exports\stores_demo.exp**. Thus, the data files (the **.unl** files) are stored in **D:\work\exports\stores_demo.exp**, and the schema file is **D:\work\exports\stores_demo.exp\stores_demo.sql**.

1. Change the name of the **.exp** directory. That is, change **D:\work\exports\stores_demo.exp** to **D:\work\exports\newname.exp**.
2. Change the name of the schema file. That is, change **D:\work\exports\stores_demo.exp\stores_demo.sql** to **D:\work\exports\stores_demo.exp\newname.sql**. Do not change the names of the **.unl** files.
3. Import the database with the following command:

```
dbimport -i D:\work\exports
```



IDS 2000

IUS

WIN NT



Simple Large Objects

When **dbimport**, **dbexport**, and DB-Access process simple-large-object data, they create temporary files for that data. Before you export or import data from tables that contain simple large objects, you must have one of the following items:

- A **\tmp** directory on your currently active drive
- The **DBTEMP** environment variable set to point to a directory that is available for temporary storage of the simple large objects

Windows NT sets the **TMP** and **TEMP** environment variables in the command prompt sessions, by default. However, if the **TMP**, **TEMP**, and **DBTEMP** environment variables are not set, **dbimport** places the temporary files for the simple large objects in the **\tmp** directory. ♦

Warning: If a table has a CLOB or BLOB in a column, you cannot use **dbexport** to export the table to a tape. If a table has a user-defined type in a column, using **dbexport** to export the table to a tape might yield unpredictable results, depending on the export function of the user-defined type. Exported CLOB sizes are stored in hex format in the unload file.

Database Locale Changes

You can use **dbimport** to change the locale of a database.

To change the locale of a database

1. Set the **DB_LOCALE** environment variable to the name of the current database locale.
2. Run **dbexport** on the database.
3. Use the **DROP DATABASE** statement to drop the database that has the current locale name.
4. Set the **DB_LOCALE** environment variable to the desired database locale for the database.
5. Run **dbimport** to create a new database with the desired locale and import the data into this database.

Using the dbload Utility

In This Chapter	21-3
Syntax of the dbload Command	21-4
Command File for dbload	21-9
Command File to Load Complex Data Types	21-21

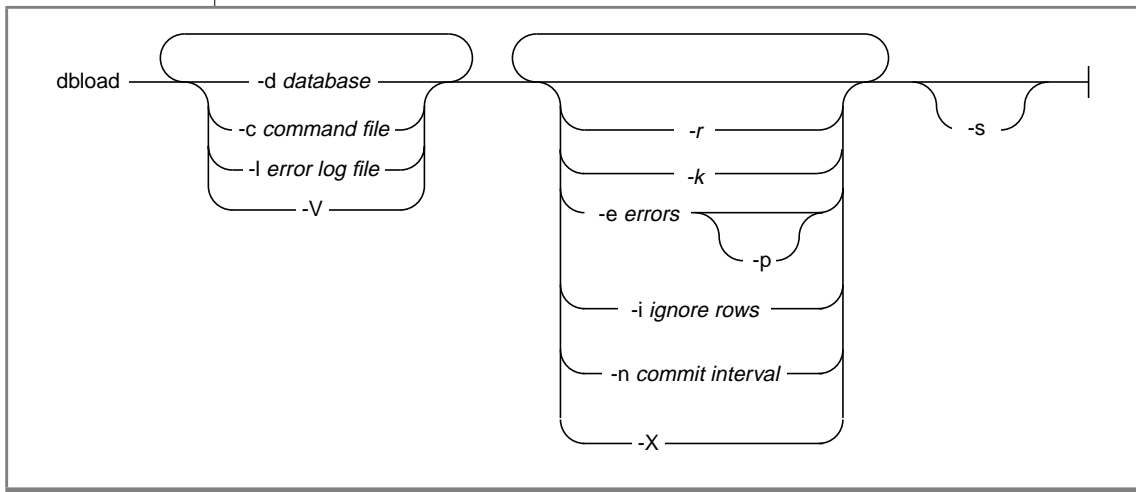
In This Chapter

This chapter describes the **dbload** utility and how to use it. You can use **dbload** with the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE
- OnLine

The **dbload** utility loads data into databases or tables that Informix products created. It transfers data from one or more text files into one or more existing tables. This utility supports new data types in Dynamic Server 2000 and Universal Server.

Syntax of the dbload Command



Element	Purpose	Key Considerations
-c command file	Specifies the filename or pathname of a dbload command file.	References: For information about building the command file, refer to “Command File for dbload” on page 21-9 .
-d database	Specifies the name of the database to receive the data.	Additional Information: If you want to use more than the simple name of the database, refer to the Database Name section of the <i>Informix Guide to SQL: Syntax</i> .
-e errors	Specifies the number of bad rows that dbload reads before terminating. The default value for <i>errors</i> is 10.	References: For more information, refer to “Bad-Row Limit” on page 21-6 .
-i ignore rows	Specifies the number of rows to ignore in the input file.	References: For more information, refer to “Rows to Ignore” on page 21-6 .
-k	Instructs dbload to lock the tables listed in the command file in exclusive mode during the load operation.	References: For more information, refer to “Table Locking” on page 21-6 . Restrictions: You cannot use the -k option with the -r option because the -r option specifies that no tables are locked during the load operation.

(1 of 2)

Element	Purpose	Key Considerations
-l <i>error log file</i>	Specifies the filename or pathname of an error log file.	<p>Restrictions: If you specify an existing file, its contents are overwritten. If you specify a file that does not exist, dbload creates the file.</p> <p>Additional Information: The error log file stores diagnostic information and any input file rows that dbload cannot insert into the database.</p>
-n <i>commit interval</i>	Specifies the commit interval in number of rows. The default interval is 100 rows.	<p>Additional Information: If your database supports transactions, dbload commits a transaction after the specified number of new rows is read and inserted. A message appears after each commit.</p> <p>References: For information about transactions, see the <i>Informix Guide to SQL: Tutorial</i>.</p>
-p	Prompts for instructions if the number of bad rows exceeds the limit.	<p>References: For more information, refer to “Bad-Row Limit” on page 21-6.</p>
-r	Prevents dbload from locking the tables during a load, thus enabling other users to update data in the table during the load.	<p>Additional Information: For more information, refer to “Table Locking” on page 21-6.</p> <p>Restrictions: You cannot use the -r option with the -k option because the -r option specifies that the tables are not locked during the load operation while the -k option specifies that the tables are locked in exclusive mode.</p>
-s	Checks the syntax of the statements in the command file without inserting data.	<p>Additional Information: The standard output displays the command file with any errors marked where they are found.</p>
-V	Displays product version information.	
-X	Recognizes HEX binary data in character fields.	

(2 of 2)



Tip: If you specify part (but not all) of the required information, **dbload** prompts you for additional specifications. The database name, command file, and error log file are all required. If you are missing all three options, you receive an error message.

Table Locking

If you do not specify the **-k** option, the tables specified in the command file are locked in shared mode. When tables are locked in shared mode, the database server still has to acquire exclusive row or page locks when it inserts rows into the table.

When you specify the **-k** option, the database server places an exclusive lock on the entire table. The **-k** option increases performance for large loads because the database server does not have to acquire exclusive locks on rows or pages as it inserts rows during the load operation.

If you do not specify the **-r** option, the tables specified in the command file are locked during loading so that other users cannot update data in the table. Table locking reduces the number of locks needed during the load but reduces concurrency. If you are planning to load a large number of rows, use table locking and load during nonpeak hours.

To override this default lock mode, specify the **-k** option. The **-k** option instructs **dbload** to lock the tables in exclusive mode rather than shared mode during the load operation.

Rows to Ignore

The **-i** option instructs **dbload** to read and ignore the specified number of new-line characters in the input file before it begins to process. This option is useful if your most recent **dbload** session ended prematurely. For example, if **dbload** ends after it inserts 240 lines of input, you can begin to load again at line 241 if you set *number rows ignore* to 240. It is also useful if header information in the input file precedes the data records.

Bad-Row Limit

The **-e** option lets you specify how many bad rows to allow before **dbload** terminates.

If you set *errors* to a positive integer, **dbload** terminates when it reads (*errors* + 1) bad rows. If you set *errors* to zero, **dbload** terminates when it reads the first bad row.

If **dbload** exceeds the bad-row limit and the **-p** option is specified, **dbload** prompts you for instructions before it terminates. The prompt asks whether you want to roll back or to commit all rows that were inserted since the last transaction.

If **dbload** exceeds the bad-row limit and the **-p** option is not specified, **dbload** commits all rows that were inserted since the last transaction.

Guidelines for Using dbload

This section includes the following guidelines for using the dbload utility:

- Termination of dbload
- Network names
- Simple large objects
- Indexes
- Delimited identifiers
- SE Example

Termination of dbload

If you press the INTERRUPT key, **dbload** terminates and discards any new rows that were inserted but not yet committed to the database (if the database has transactions).

Network Names

If you are on a network, include the database server name and directory path with the database name to specify a database on another database server or coserver.

Simple Large Objects

You can load simple large objects with the **dbload** utility as long as the simple large objects are in text files.

Indexes

The presence of indexes greatly affects the speed with which the **dbload** utility loads data. For best performance, drop any indexes on the tables that receive the data before you run **dbload**. You can create new indexes after **dbload** has finished.

Delimited Identifiers

You can use delimited identifiers with the **dbload** utility. The utility detects database objects that are keywords, mixed case, or have special characters, and places double quotes around them.

If your most recent **dbload** session ended prematurely, specify the starting line number in the command-line syntax to resume loading with the next record in the file.

SE Example

The following command loads data into the **stores_demo** database in the **turku** directory on the SE database server **finland**:

```
dbload -d //finland/turku/stores_demo -c commands -l errlog
```

Command File for dbload

Before you use **dbload**, you must create a command file that names the input data files and the tables that receive the data. The command file maps fields from one or more input files into columns of one or more tables within your database.

The command file contains only FILE and INSERT statements. Each FILE statement names an input data file. The FILE statement also defines the data fields from the input file that are inserted into the table. Each INSERT statement names a table to receive the data. The INSERT statement also defines how **dbload** places the data that is described in the FILE statement into the table columns.

Within the command file, the FILE statement can appear in the following forms:

- Delimiter form
- Character-position form

The FILE statement has a size limit of 4,096 bytes.

Use the delimiter form of the FILE statement when every field in the input data row uses the same delimiter and every row ends with a new-line character. This format is typical of data rows with variable-length fields. You can also use the delimiter form of the FILE statement with fixed-length fields as long as the data rows meet the delimiter and new-line requirements. The delimiter form of the FILE and INSERT statements is easier to use than the character-position form.

Use the character-position form of the FILE statement when you cannot rely on delimiters and you need to identify the input data fields by character position within the input row. For example, use this form to indicate that the first input data field begins at character position 1 and continues until character position 20. You can also use this form if you must translate a character string into a null value. For example, if your input data file uses a sequence of blanks to indicate a null value, you must use this form if you want to instruct **dbload** to substitute null at every occurrence of the blank-character string.

You can use both forms of the FILE statement in a single command file. However, for clarity, the two forms are described separately in the following sections.

FILE and INSERT Statements: Delimiter Form

The following example of a **dbload** command file illustrates a simple delimiter form of the FILE and INSERT statements. The example is based on the **stores_demo** database. An UNLOAD statement created the three input data files, **stock.unl**, **customer.unl**, and **manufact.unl**. To see the **.unl** input data files, refer to the directory **\$INFORMIXDIR/demo/prod_name** (UNIX) or **%INFORMIXDIR%\demo\prod_name** (Windows NT).

```
FILE stock.unl DELIMITER '|' 6;
INSERT INTO stock;
FILE customer.unl DELIMITER '|' 10;
INSERT INTO customer;
FILE manufact.unl DELIMITER '|' 3;
INSERT INTO manufact;
```

Syntax for the Delimiter Form

The following diagram shows the syntax of the delimiter FILE statement.

FILE — *filename* — DELIMITER — 'c' — *nfields* —

Element	Purpose	Key Considerations
<i>c</i>	Defines the field delimiter for the specific input file.	Restrictions: If the delimiter specified by <i>c</i> appears as a literal character anywhere in the input file, the character must be preceded with a backslash (\) in the input file. For example, if the value of <i>c</i> is specified as a square bracket ([), you must place a backslash before any literal square bracket that appears in the input file. Similarly, you must precede any backslash that appears in the input file with an additional backslash.
<i>filename</i>	Specifies the input file.	None.
<i>nfields</i>	Indicates the number of fields in each data row.	None.

The **dbload** utility assigns the sequential names **f01**, **f02**, **f03**, and so on to fields in the input file. You cannot see these names, but if you refer to these fields to specify a value list in an associated INSERT statement, you must use the **f01**, **f02**, **f03** format. For details, refer to [“How to Write a dbload Command File in Delimiter Form” on page 21-13](#).

Two consecutive delimiters define a null field. As a precaution, you can place a delimiter immediately before the new-line character that marks the end of each data row. If the last field of a data row has data, you must use a delimiter. If you omit this delimiter, an error results whenever the last field of a data row is not empty.

Inserted data types correspond to the explicit or default column list. If the data field width is different from its corresponding character column width, the data is made to fit. That is, inserted values are padded with blanks if the data is not wide enough for the column or truncated if the data is too wide for the column.

If the number of columns named is fewer than the number of columns in the table, **dbload** inserts the default value that was specified when the table was created for the unnamed columns. If no default value is specified, **dbload** attempts to insert a null value. If the attempt violates a not null restriction or a unique constraint, the insert fails, and an error message is returned.

If the INSERT statement omits the column names, the default INSERT specification is every column in the named table. If the INSERT statement omits the VALUES clause, the default INSERT specification is every field of the previous FILE statement.

An error results if the number of column names listed (or implied by default) does not match the number of values listed (or implied by default).

The syntax of **dbload** INSERT statements resembles INSERT statements in SQL, except that in **dbload**, INSERT statements cannot incorporate SELECT statements.



Warning: Do not use the *CURRENT*, *TODAY*, and *USER* keywords of the *INSERT INTO* statement in a **dbload** command file; they are not supported in the **dbload** command file. These keywords are supported in SQL only.

For example, the following **dbload** command is not supported:

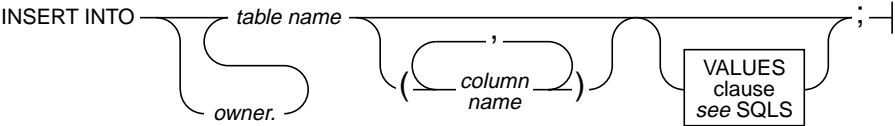
```
FILE "testtbl2.unl" DELIMITER '|' 1;  
INSERT INTO testtbl  
  (testuser, testtime, testfield)  
VALUES  
  ('kae', CURRENT, f01);
```

Load the existing data first and then write an SQL query to insert or update the data with the current time, date, or user login. You could write the following SQL statement:

```
INSERT INTO testtbl  
  (testuser, testtime, testfield)  
VALUES  
  ('kae', CURRENT, f01);
```

The **CURRENT** keyword returns the system date and time. The **TODAY** keyword returns the system date. The **USER** keyword returns the user login name.

The following diagram shows the syntax of the **dbload** INSERT statement for delimiter form.



Element	Purpose	Key Considerations
<i>column name</i>	Specifies the column that receives the new data.	None.
<i>owner.</i>	Specifies the user name of the table owner.	None.
<i>table name</i>	Specifies the table that receives the new data.	None.

Users who execute **dbload** with this command file must have the Insert privilege on the named table.

How to Write a dbload Command File in Delimiter Form

The first FILE AND INSERT statement set in the delimiter example on [page 21-10](#) is repeated in the following example:

```
FILE stock.unl DELIMITER '|' 6;
INSERT INTO stock;
```

The FILE statement describes the **stock.unl** data rows as composed of six fields, each separated by a vertical bar (|) as the delimiter. Two consecutive delimiters define a null field. As a precaution, you can place a delimiter immediately before the new line character that marks the end of each data row. If the last field of a data row has data, you must use a delimiter. If you omit this delimiter, an error results.

Compare the FILE statement with the data rows in the following example, which appear in the input file **stock.unl**. (Because the last field is not followed by a delimiter, an error results if any data row ends with an empty field.)

```
1|SMT|baseball gloves|450.00|case|10 gloves/case
2|HRO|baseball|126.00|case|24/case
3|SHK|baseball bat|240.00|case|12/case
```

The example INSERT statement contains only the required elements. Because the column list is omitted, the INSERT statement implies that values are to be inserted into every field in the **stock** table. Because the VALUES clause is omitted, the INSERT statement implies that the input values for every field are defined in the most-recent FILE statement. This INSERT statement is valid because the **stock** table contains six fields, which is the same number of values that the FILE statement defines. The following example shows the first data row that is inserted into **stock** from this INSERT statement.

Field	Column	Value
f01	stock_num	1
f02	manu_code	SMT
f03	description	baseball gloves

(1 of 2)

Field	Column	Value
f04	unit_price	450.00
f05	unit	case
f06	unit_descr	10 gloves/case

(2 of 2)

The FILE and INSERT statement in the following example illustrates a more complex INSERT statement syntax:

```
FILE stock.unl DELIMITER '|' 6;  
INSERT INTO new_stock (col1, col2, col3, col5, col6)  
VALUES (f01, f03, f02, f05, 'autographed');
```

In this example, the VALUES clause uses the field names that **dbload** assigns automatically. You must reference the automatically assigned field names with the letter **f** followed by a number: **f01**, **f02**, **f10**, **f100**, **f999**, **f1000**, and so on. All other formats are incorrect.

Tip: The first nine fields must include a zero: f01, f02, ..., f09.

The user changed the column names, the order of the data, and the meaning of **col6** in the new **stock** table. Because the fourth column in **new_stock** (**col4**) is not named in the column list, the new data row contains a null in the **col4** position (assuming that the column permits nulls). If no default is specified for **col4**, the inserted value is null.

The following table shows the first data row that is inserted into **new_stock** from this INSERT statement.

Column	Value
col1	1
col2	baseball gloves
col3	SMT
col4	null
col5	case
col6	autographed



FILE and INSERT Statements: Character-Position Form

The examples in this section are based on an input data file, **cust_loc_data**, that contains the last four columns (**city**, **state**, **zipcode**, and **phone**) of the **customer** table. Fields in the input file are padded with blanks to create data rows in which the location of data fields and the number of characters are the same across all rows. The definitions for these fields are CHAR(15), CHAR(2), CHAR(5), and CHAR(12), respectively. [Figure 21-1](#) displays the character positions and five example data rows from the **cust_loc_data** file.

1234567890123456789012345678901234

Sunnyvale	CA94086408-789-8075
Denver	C080219303-936-7731
Blue Island	NY60406312-944-5691
Brighton	MA02135617-232-4159
Tempe	AZ85253xxx-xxx-xxxx

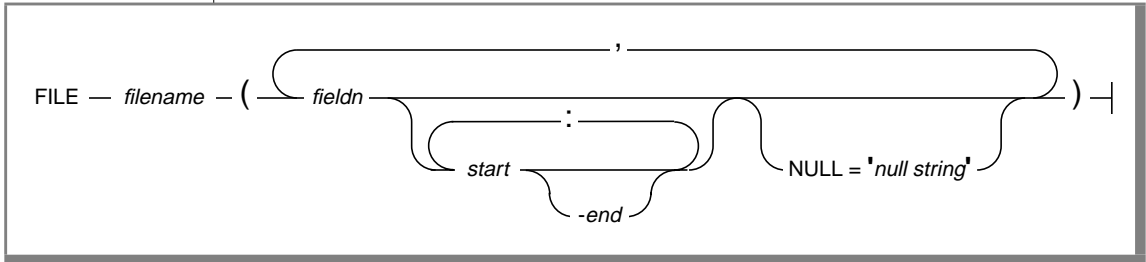
Figure 21-1
A Sample Data File

The following example of a **dbload** command file illustrates the character-position form of the FILE and INSERT statements. The example includes two new tables, **cust_address** and **cust_sort**, to receive the data. For the purpose of this example, **cust_address** contains four columns, the second of which is omitted from the column list. The **cust_sort** table contains two columns.

```
FILE cust_loc_data
  (city 1-15,
   state 16-17,
   area_cd 23-25 NULL = 'xxx',
   phone 23-34 NULL = 'xxx-xxx-xxxx',
   zip 18-22,
   state_area 16-17 : 23-25);
INSERT INTO cust_address (col1, col3, col4)
  VALUES (city, state, zip);
INSERT INTO cust_sort
  VALUES (area_cd, zip);
```

Syntax for the Character-Position Form

The following diagram shows the syntax of the character-position FILE statement.



Element	Purpose	Key Considerations
<i>-end</i>	Indicates the character position within a data row that ends a range of character positions.	Restrictions: A hyphen must precede the end value.
<i>fieldn</i>	Assigns a name to the data field that you are defining with the range of character positions.	None.
<i>filename</i>	Specifies the name of the input file.	None.
<i>null string</i>	Specifies the data value for which dbload should substitute a null.	Restrictions: Must be a quoted string.
<i>start</i>	Indicates the character position within a data row that starts a range of character positions. If you use <i>start</i> without <i>end</i> , it represents a single character.	None.

You can repeat the same character position in a data-field definition or in different fields.

The *null string* scope of reference is the data field for which you define it. You can define an explicit null string for each field that allows null entries.

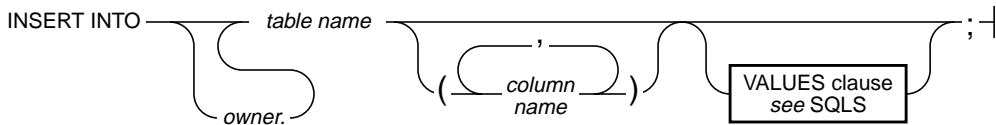
Inserted data types correspond to the explicit or default column list. If the data-field width is different from its corresponding character column, inserted values are padded with blanks if the column is wider or are truncated if the field is wider.

If the number of columns named is fewer than the number of columns in the table, **dbload** inserts the default value that is specified for the unnamed columns. If no default value is specified, **dbload** attempts to insert a null value. If the attempt violates a not null restriction or a unique constraint, the insert fails, and an error message is returned.

If the INSERT statement omits the column names, the default INSERT specification is every column in the named table. If the INSERT statement omits the VALUES clause, the default INSERT specification is every field of the previous FILE statement.

An error results if the number of column names listed (or implied by default) does not match the number of values listed (or implied by default).

The syntax of **dbload** INSERT statements resembles INSERT statements in SQL, except that in **dbload**, INSERT statements cannot incorporate SELECT statements. The following diagram shows the syntax of the **dbload** INSERT statement for character-position form.



Element	Purpose	Key Considerations
<i>column name</i>	Specifies the column that receives the new data.	None.
<i>table name</i>	Specifies the table that receives the new data.	None.
<i>owner.</i>	Specifies the user name of the table owner.	None.

The syntax for character-position form is identical to the syntax for delimiter form.

The user who executes **dbload** with this command file must have the Insert privilege on the named table.

SE

In SE, the **dbload** utility recognizes valid SE table references, including owner designations. That is, the owner name can precede the table name but the database server name or the database name cannot precede the table name. Valid table-name syntax is defined in detail in the *Informix Guide to SQL: Syntax*. ♦

How to Write a dbload Command File in Character-Position Form

The first FILE AND INSERT statement set in the character-position example on [page 21-15](#) is repeated in the following example:

```
FILE cust_loc_data
(city 1-15,
 state 16-17,
 area_cd 23-25 NULL = 'xxx',
 phone 23-34 NULL = 'xxx-xxx-xxxx',
 zip 18-22,
 state_area 16-17 : 23-25);
INSERT INTO cust_address (col1, col3, col4)
VALUES (city, state, zip);
```

The FILE statement defines six data fields from the **cust_loc_data** table data rows. The statement names the fields and uses character positions to define the length of each field. Compare the FILE statement in the preceding example with the data rows in [Figure 21-2](#).

1	2	3	
1234567890123456789012345678901234			
Sunnyvale++++++CA94086408-789-8075			—Data row 1
Tempe++++++AZ85253xxx-xxx-xxxx			—Data row 2

Figure 21-2
A Sample Data File

The FILE statement defines the following data fields, which are derived from the data rows in [Figure 21-2](#).

Column	Values from Data Row 1	Values from Data Row 2
city	Sunnyvale++++++	Tempe+++++++
state	CA	AZ
area_cd	408	null
phone	408-789-8075	null
zip	94086	85253
state_area	CA408	AZxxx

The null strings that are defined for the **phone** and **area_cd** fields generate the null values in those columns but do not affect the values that are stored in the **state_area** column.

The INSERT statement uses the field names and values that are derived from the FILE statement as the value-list input. Consider the following INSERT statement:

```
INSERT INTO cust_address (col1, col3, col4)
VALUES (city, state, zip);
```

The INSERT statement uses the data in [Figure 21-2](#) and the FILE statement on [page 21-18](#) to put the following information into the **cust_address** table.

Column	Values from Data Row 1	Values from Data Row 2
col1	Sunnyvale++++++	Tempe+++++++
col2	null	null
col3	CA	AZ
col4	94086	85253

Because the second column (**col2**) in **cust_address** is not named, the new data row contains a null (assuming that the column permits nulls).

Consider the following INSERT statement:

```
INSERT INTO cust_sort
VALUES (area_cd, zip);
```

This INSERT statement inserts the following data rows into the **cust_sort** table.

Column	Values from Data Row 1	Values from Data Row 2
col1	408	NULL
col2	94086	85253

Because no column list is provided, **dbload** reads the names of all the columns in **cust_sort** from the system catalog. (You cannot insert data into a temporary table because temporary tables are not entered into the system catalog.) Field names from the previous FILE statement specify the values to load into each column. You do not need one FILE statement for each INSERT statement.

IDS 2000

IUS

Command File to Load Complex Data Types

This section describes how to write **dbload** command files that load columns that contain complex data types into tables. The examples cover how to use **dbload** with named row types, unnamed row types, sets, and lists.

Using dbload with Named Row Types

The procedure for how to use **dbload** with named row types is somewhat different than for other complex data types because named row types are actually user-defined data types. In fact, you can follow these steps for any user-defined data type.

This example uses a table **person** that contains one column with a named row type. The **person_t** named row type contains six fields: **name**, **address**, **city**, **state**, **zip**, and **bdate**.

The following syntax shows how to create the named row type and the table used in this example:

```
CREATE ROW TYPE person_t
(
    name VARCHAR(30) NOT NULL,
    address VARCHAR(20),
    city VARCHAR(20),
    state CHAR(2),
    zip VARCHAR(9),
    bdate DATE
);
CREATE TABLE person OF TYPE person_t;
```

To load the data, follow these steps:

1. Use the **UNLOAD** statement to unload the table to an input file. In this example, the input file sees the named row type as six separate fields:

```
Brown, James|13 First St.|San Francisco|CA|94070|01/04/1940|
Karen Smith|5820 Easy Ave #100|Fremont|CA|94502|01/13/1983|
```

2. Use the **dbschema** utility to capture the schema of the table and the row type. You must use the **dbschema -u** option to pick up the named row type.

```
dbschema -d stores_demo -u person_t > schema.sql
dbschema -d stores_demo -t person > schema.sql
```

3. Use DB-Access to re-create the **person** table in the new database. For the detailed steps, see “[DB-Access Input from dbschema Output](#)” on page 22-22.
4. Create the **dbload** command file. This **dbload** command file inserts two rows into the **person** table in the new database.

```
FILE person.unl DELIMITER '|' 6;  
INSERT INTO person;
```

This **dbload** example shows how to insert new data rows into the **person** table. The number of rows in the INSERT statement and the **dbload** command file must match:

```
FILE person.unl DELIMITER '|' 6;  
INSERT INTO person  
VALUES ('Jones, Richard', '95 East Ave.', 'Philadelphia', 'PA',  
'19115',  
'03/15/97');
```

5. Execute the **dbload** command:

```
dbload -d newdb -c uds_command -l errlog
```

Tip: To find the number of fields in an unloaded table that contains a named row type, count the number of fields between each vertical bar (|) delimiter.



Using dbload with Unnamed Row Types

You can use **dbload** with unnamed row types. In the following example, the **devtest** table contains two columns with unnamed row types, **s_name** and **s_address**. The **s_name** column contains three fields: **f_name**, **m_init**, and **l_name**. The **s_address** column contains four fields: **street**, **city**, **state**, and **zip**.

```
CREATE TABLE devtest  
(  
    s_name ROW(f_name varchar(20), m_init char(1), l_name  
    varchar(20) not null),  
    s_address ROW(street varchar(20), city varchar(20), state  
    char(20), zip varchar(9))  
);
```

The data from the **devtest** table is unloaded into the **devtest.unl** file. Each data row contains two delimited fields, one for each unnamed row type. The ROW constructor precedes each unnamed row type, as follows:

```
ROW('Jim','K','Johnson')|ROW('10 Grove St.','Eldorado','CA','94108')|  
ROW('Candy','S','Cane')|ROW('7 Willy Wonka  
Ave.','Hershey','PA','17033')|
```

This **dbload** example shows how to insert data that contains unnamed row types into the **devtest** table. Put double quotes around each unnamed row type or the insert will not work.

```
FILE devtest.unl DELIMITER '|' 2;
INSERT INTO devtest (s_name, s_address)
VALUES ("row('Craig', 'X', 'Smith')",
        "row('1200 Cheese Ave.', 'Rainy City', 'OR', '97200')");
```

Using dbload with Collection Data Types

You can use **dbload** with collection data types such as SET, LIST, and MULTiset.

SET Data Type Example

In a SET, each element is unique, and no nulls are allowed. The numbers of elements in a SET can vary. The following statement creates a table in which the **children** column is defined as a SET:

```
CREATE TABLE employee
(
    name char(30),
    address char(40),
    children SET (varchar(30) NOT NULL)
);
```

The data from the **employee** table is unloaded into the **employee.unl** file. Each data row contains four delimited fields. The first set contains three elements (**Karen**, **Lauren**, and **Andrea**) while the second set contains four elements. The SET constructor precedes each SET data row.

```
Muriel|5555 SW Merry
SailingDr.|02/06/1926|SET{'Karen','Lauren','Andrea'}|
Larry|1234 Indian Lane|07/31/1927|SET{'Martha','Melissa','Craig','Larry'}|
```

This **dbload** example shows how to insert data that contains SET data types into the **employee** table in the new database. Put double quotes around each SET data type or the insert does not work.

```
FILE employee.unl DELIMITER '|' 4;
INSERT INTO employee
VALUES ('Marvin', '10734 Pardee', '06/17/27', "SET{'Joe', 'Ann'}");
```

LIST Data Type Example

A list is an ordered collection of elements that allows duplicate values. The following statement creates a table in which the **month_sales** column is defined as a LIST:

```
CREATE TABLE sales_person
(
    name CHAR(30),
    month_sales LIST(MONEY NOT NULL)
);
```

The data from the **sales_person** table is unloaded into the **sales.unl** file. Each data row contains two delimited fields, as follows:

```
Jane Doe|LIST{'4.00','20.45','000.99'}|
Big Earner|LIST{'0000.00','00000.00','999.99'}|
```

This **dbload** example shows how to insert data that contains LIST data types into the **sales_person** table in the new database. Put double quotes around each LIST data type or else the insert does not work.

```
FILE sales_person.unl DELIMITER '|' 2;
INSERT INTO sales_person
VALUES ('Jenny Chow', "list{587900, 600000}");
```

You can load multisets in a similar manner.

Using dbload with Other Data Types

You can use **dbload** with the following data types:

- A BLOB or CLOB
- A SET inside a ROW type

The **dbload** utility does not work with the following data types:

- A CLOB or BLOB inside a ROW type
- A ROW type inside a SET



Warning: All the load utilities (**dbexport**, **dbimport**, **load**, **unload**, and **dbload**) rely on an export and import function. If you do not define this function when you write a user-defined type, you cannot use these utilities.

Loading a new data type inside another data type can cause problems if the representation of the data contains handles. If a string represents the data, you should be able to load it.

Using the dbschema Utility

In This Chapter	22-3
Syntax of the dbschema Command	22-4
Database Schema Creation	22-6
Server-Specific Information	22-8
User-Defined and Complex Data Types	22-9
Synonym Creation	22-11
Privileges	22-12
Table, View, or Procedure Creation	22-14
Table Information	22-16
Role Creation	22-17
Distribution Information for Tables	22-18
DB-Access Input from dbschema Output	22-22

In This Chapter

This chapter describes the **dbschema** utility and how to use it. You can use **dbschema** with the following database servers:

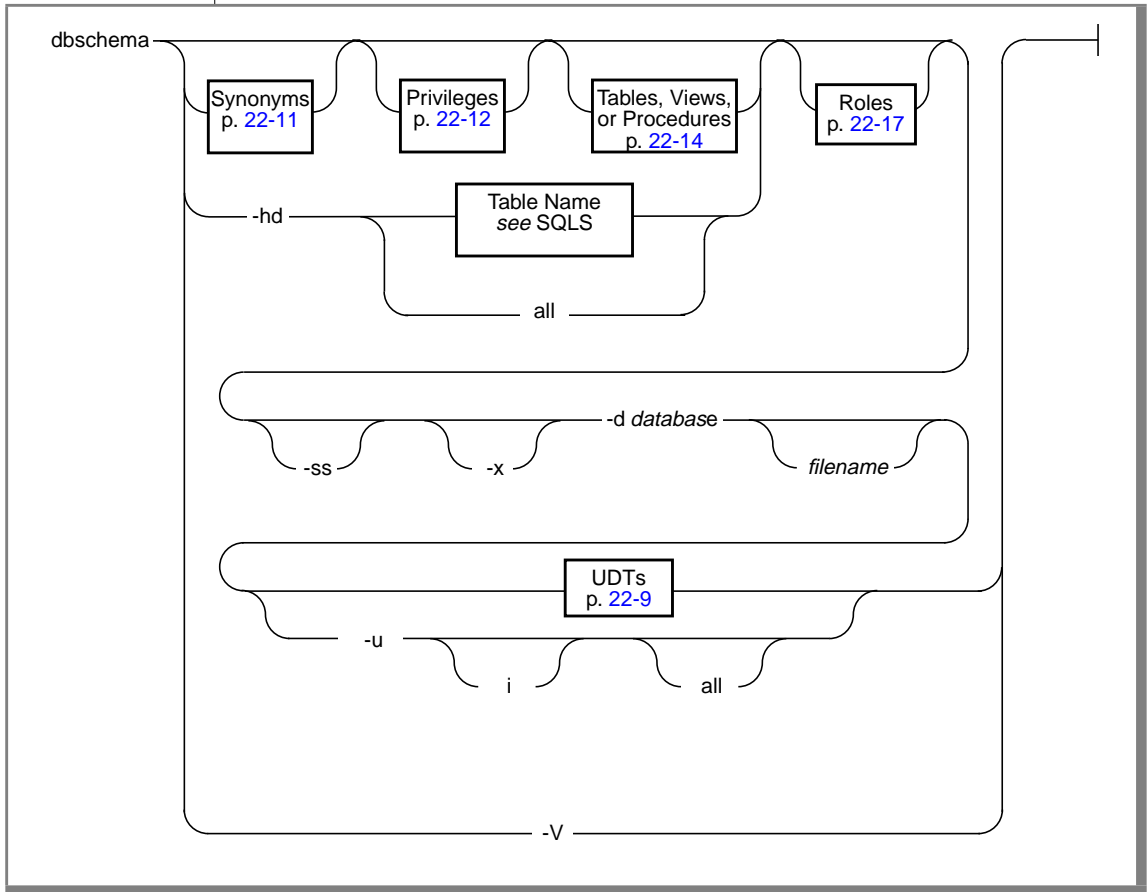
- Dynamic Server 2000
- Universal Server
- Extended Parallel Server
- Dynamic Server with AD and XP Options
- OnLine XPS
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE
- OnLine

The **dbschema** utility prints SQL statements necessary to replicate a specified table, view, or database. It also shows the distributions that UPDATE STATISTICS creates.

You can use the **dbschema** utility for the following purposes:

- To display the SQL statements (the *schema*) that are required to replicate a database or a specific table, view, or procedure
- To display the schema for the Information Schema views
- To display the distribution information that is stored for one or more tables in the database
- To display information on user-defined data types and row types

Syntax of the dbschema Command



Element	Purpose	Key Considerations
-d <i>database/co-server</i>	Specifies the database or co-server to which the schema applies. The database can be on a remote database server.	References: If you want to use more than the simple name of the database, refer to the Database Name section of the <i>Informix Guide to SQL: Syntax</i> .
-hd	Displays the distribution as data values.	References: For more information, refer to “Distribution Information for Tables” on page 22-18 .
-ss	Generates server-specific information.	Restrictions: This option is ignored if no table schema is generated. References: For more information, refer to “Server-Specific Information” on page 22-8 .
-u	Prints the definitions of user-defined data types.	References: For more information, refer to “User-Defined and Complex Data Types” on page 22-9 .
-ui	Prints the definitions of user-defined data types, including type inheritance.	References: For more information, refer to “User-Defined and Complex Data Types” on page 22-9 .
-V	Displays product version information.	None.
-x	Expands dbslice names into dbspace name lists in -ss output.	None.
all	Directs dbschema to include all the tables in the database in the display of distributions.	None.
<i>filename</i>	Specifies the filename to contain the dbschema output.	Additional Information: If you do not supply a <i>filename</i> , dbschema sends output to the screen. If you do supply a <i>filename</i> , dbschema creates a file to contain the dbschema output and gives it the name you specify.

You must be the DBA or have the Connect or Resource privilege to the database before you can run **dbschema** on it.

Database Schema Creation

You can create the schema for an entire database or for a portion of the database. The options for **dbschema** allow you to perform the following actions:

- Display CREATE SYNONYM statements by owner, for a specific table or for the entire database.
- Display the CREATE TABLE, CREATE VIEW, CREATE FUNCTION, or CREATE PROCEDURE statements for a specific table or for the entire database.
- Display all GRANT privilege statements that affect a specified user or that affect all users for a database or a specific table. The user can be either a user name or role name.
- Display user-defined and row data types with or without type inheritance. ♦

When you use **dbschema** and specify only the database name, it is equivalent to using **dbschema** with all its options (except for the **-hd** and **-ss** options). In addition, if Information Schema views were created for the database, this schema is shown. For example, the following two commands are equivalent:

```
dbschema -d stores_demo
dbschema -s all -p all -t all -f all -d stores_demo
```

The SERIAL fields included in CREATE TABLE statements that **dbschema** displays do not specify a starting value. New SERIAL fields created with the schema file have a starting value of 1, regardless of their starting value in the original database. If this value is not acceptable, you must modify the schema file.

UNIX

Creating Schemas for Databases Across a Network

You can specify a database on any accessible non-SE Informix database servers with the **-d** database syntax. The following command displays the schema for the **stores_demo** database on the **finland** database server on the UNIX system console:

```
dbschema -d //finland/stores_demo
```



SE

To specify a database on an SE database server, include the database server name and directory path with the database name. The command in the following example displays the schema for the **stores_demo** database in the **turku** directory on the **finland** database server on the system console:

```
dbschema -d //finland/turku/stores_demo
```



Changing the Owner of an Object

The **dbschema** utility uses the *owner.object* convention when it generates any CREATE TABLE, CREATE INDEX, CREATE SYNONYM, CREATE VIEW, CREATE PROCEDURE, CREATE FUNCTION, or GRANT statements, and when it reproduces any unique, referential, or check constraints. As a result, if you use the **dbschema** output to create a new object (table, index, view, procedure, constraint, or synonym), the owner of the original object owns the new object. If you want to change the owner of the new object, you must edit the **dbschema** output before you run it as an SQL script.

You can use the output of **dbschema** to create a new function if you also specify the *pathname* to a file in which compile-time warnings are stored. This pathname is displayed in the **dbschema** output.

For more information about the CREATE TABLE, CREATE INDEX, CREATE SYNONYM, CREATE VIEW, CREATE PROCEDURE, CREATE FUNCTION, and GRANT statements, see the *Informix Guide to SQL: Syntax*.

Server-Specific Information

The **-ss** option generates server-specific information. In all Informix database servers except SE, the **-ss** option always generates the lock mode, extent sizes, and the dbspace name if the dbspace name is different from the database dbspace. In addition, if tables are fragmented, the **-ss** option displays information about the fragmentation strategy.

When you specify the **dbschema -ss** option, the output also displays any GRANT FRAGMENT statements that are issued for a particular user or in the entire schema.

You can use the **-x** option to expand dbslice names into dbspace name lists in the **-ss** output.



Important: Use the **dbschema -ss** option to obtain information specific to a database server, including fragmentation and storage options.

For more information about fragment-level authority, see the GRANT FRAGMENT and REVOKE FRAGMENT statements in the *Informix Guide to SQL: Syntax*.

SE

In SE, the **-ss** option generates the pathname where the table was created if the table is not in the database directory. ♦

IDS 2000

IUS

User-Defined and Complex Data Types

When you specify the **dbschema -u** option, the output displays the definitions of any user-defined and complex data types that the database contains. The suboption **i** lets you display the type inheritance.

The following command displays all the user-defined and complex data types for the **stork** database:

```
dbschema -d stork -u all
```

Output from **dbschema** that is executed with the specified option **-u all** might appear as the following example shows:

```
create row type 'informix'.person_t
(
    name varchar(30, 10) not null,
    address varchar(20, 10),
    city varchar(20, 10),
    state char(2),
    zip integer,
    bdate date
);
create row type 'informix'.employee_t
(
    salary integer,
    manager varchar(30, 10)
) under person_t;
```

The following command displays the user-defined and complex data types, as well as their type inheritance for the **person_t** table in the **stork** database:

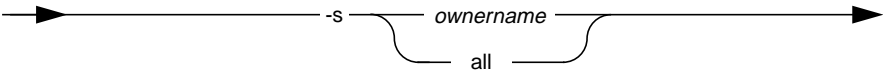
```
dbschema -d stork -ui person_t
```

Output from **dbschema** executed with the option `-ui person_t` might appear as the following example shows:

```
create row type 'informix'.person_t
(
  name varchar(30, 10) not null,
  address varchar(20, 10),
  city varchar(20, 10),
  state char(2),
  zip integer,
  bdate date
);
create row type 'informix'.employee_t
(
  salary integer,
  manager varchar(30, 10)
) under person_t;
create row type 'informix'.sales_rep_t
(
  rep_num integer,
  region_num integer,
  commission decimal(16),
  home_office boolean
) under employee_t;
```

Synonym Creation

Synonyms



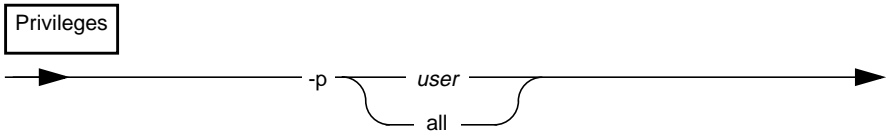
Element	Purpose	Key Considerations
-s ownername	Displays the CREATE SYNONYM statements owned by ownername.	None.
-s all	Displays all CREATE SYNONYM statements for the database, table, or view specified.	None.

Output from **dbschema** that is executed with the specified option -s alice might appear as the following example shows:

```
CREATE SYNONYM 'alice'.cust FOR 'alice'.customer
```

For more information about the CREATE SYNONYM statement, see the *Informix Guide to SQL: Syntax*.

Privileges



Element	Purpose	Key Considerations
<code>-p user</code>	Displays the GRANT statements that grant privileges to a user where <i>user</i> can be a user name or role name. Specify only one user or role.	Restriction: You cannot specify a specific list of users with the <code>-p</code> option. You can specify either one user or role, or all users and roles.
<code>-p all</code>	Displays the GRANT statements for all users for the database, table, or view specified, or to all roles for the table specified.	None.

The output also displays any GRANT FRAGMENT statements that are issued for a particular user or role or the entire schema.

Granting Privileges

In the `dbschema` output, the AS keyword indicates the grantor of a GRANT statement. The following example output indicates that **norma** issued the GRANT statement:

```
GRANT ALL ON 'tom'.customer TO 'claire' AS 'norma'
```

When the GRANT and AS keywords appear in the **dbschema** output, you might need to grant privileges before you run the **dbschema** output as an SQL script. Referring to the previous example output line, the following conditions must be true before you can run the statement as part of a script:

- User **norma** must have the Connect privilege to the database.
- User **norma** must have all privileges WITH GRANT OPTION for the table **tom.customer**.

For more information about the GRANT, GRANT FRAGMENT, and REVOKE FRAGMENT statements, see the *Informix Guide to SQL: Syntax*.

Displaying Privilege Information for a Role

A *role* is a classification with privileges on database objects granted to the role. The DBA can assign the privileges of a related work task, such as an engineer, to a role and then grant that role to users, instead of granting the same set of privileges to every user. After a role is created, the DBA can use the GRANT statement to grant the role to users or to other roles.

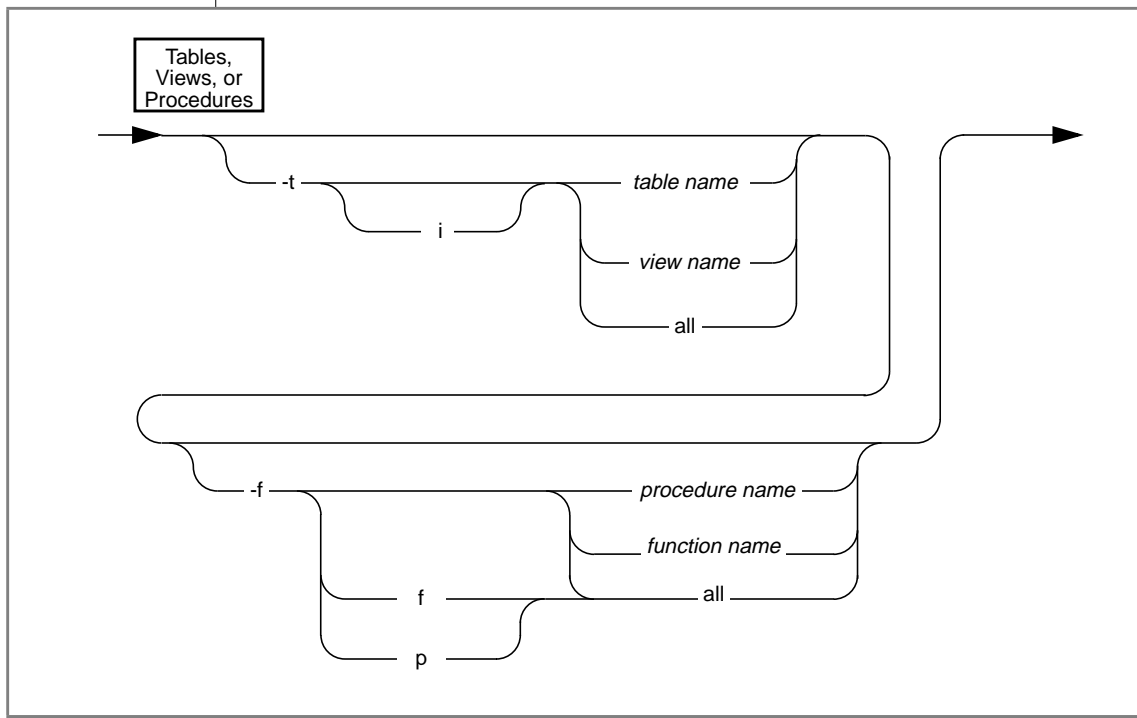
The following **dbschema** command and output show the privileges that were granted for the **calen** role:

```
sharky% dbschema -p calen -d stores_demo
```

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 7.22  
Copyright (C) Informix Software, Inc., 1984-1996  
Software Serial Number RDS#N000000
```

```
grant alter on table1 to 'calen'
```

Table, View, or Procedure Creation



Element	Purpose	Key Considerations
-f all	Limits the SQL statement output to those statements that are needed to replicate all functions and procedures.	None.
-f function name	Limits the SQL statement output to only those statements that are needed to replicate the specified function.	None.
-f procedure name	Limits the SQL statement output to only those statements that are needed to replicate the specified procedure.	None.
-ff all	Limits the SQL statement output to those statements that are needed to replicate all functions.	None.
-fp all	Limits the SQL statement output to those statements that are needed to replicate all procedures.	None.

(1 of 2)

Element	Purpose	Key Considerations
-t <i>table name</i>	Limits the SQL statement output to only those statements that are needed to replicate the specified table.	None.
-t <i>view name</i>	Limits the SQL statement output to only those statements that are needed to replicate the specified view.	None.
-t all	Includes in the SQL statement output all statements that are needed to replicate all tables and views.	None.
-ti <i>table name</i>	Includes in the SQL statement output all statements that are needed to replicate all table levels.	None.
-ti all	Includes in the SQL statement output all statements that are needed to replicate all tables and views. Functionally equivalent to -t all .	None.

(2 of 2)

For more information about the CREATE PROCEDURE and CREATE FUNCTION statements, see the *Informix Guide to SQL: Syntax*.

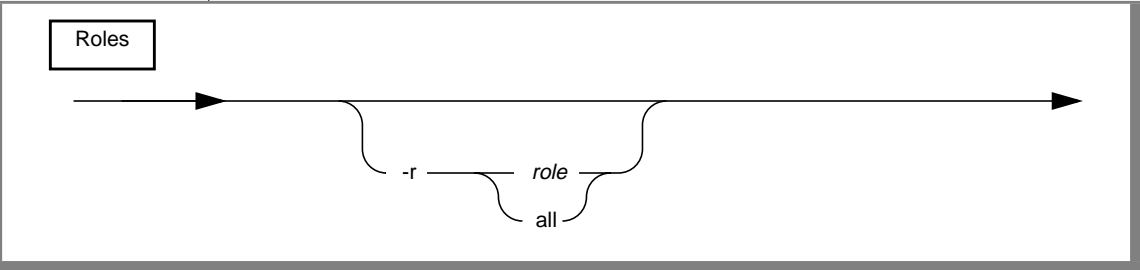
Table Information

When you use the **-ss** option, you can retrieve information about fragmented tables, the lock mode, and extent sizes.

The following **dbschema** output shows the expressions specified for fragmented table.

```
DBSCHEMA Schema Utility      INFORMIX-SQL Version 7.20.UC1
Copyright (C) Informix Software, Inc., 1984-1995
{ TABLE "sallyc".t1 row size = 8 number of columns = 1 index size = 0 }
create table "sallyc".t1
(
  c1 integer
) fragment by expression
(c1 < 100 ) in db1 ,
((c1 >= 100 ) AND (c1 < 200 ) ) in db2 ,
remainder in db4
extent size 16 next size 16 lock mode page;
revoke all on "sallyc".t1 from "public";
```


Role Creation



Element	Purpose	Key Considerations
-r role	Displays the CREATE ROLE and GRANT statements that are needed to replicate and grant the specified role.	Restriction: You cannot specify a list of users or roles with the -r option. You can specify either one role, or all roles. SE does not support the -r option.
-r all	Displays all CREATE ROLE and GRANT statements that are needed to replicate and grant all roles.	None.

The following **dbschema** command and output show that the role **calen** was created and was granted to **cathl**, **judith**, and **sallyc**:

```
sharky% dbschema -r calen -d stores_demo
```

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 7.22
Copyright (C) Informix Software, Inc., 1984-1996
Software Serial Number RDS#N000000
create role calen;

grant calen to cathl with grant option;
grant calen to judith ;
grant calen to sallyc ;
```

Distribution Information for Tables

To display the distribution information that is stored for a table in a database, use the **-hd** option with the name of the table. If you specify the ALL keyword for the table name, the distributions for all the tables in the database are displayed.

Distribution information is stored only if you have run the UPDATE STATISTICS...MEDIUM or HIGH statement for one or more columns of a table. For information about the UPDATE STATISTICS statement, refer to the *Informix Guide to SQL: Syntax*.

The output of **dbschema** for distributions is provided in the following parts:

- Distribution description
- Distribution information
- Overflow information

Each section of **dbschema** output is explained in the following sections. As an example, the discussion uses the following distribution for the fictional table called **invoices**. This table contains 165 rows, including duplicates.

You can generate the output for this discussion with a call to **dbschema** that is similar to the following example:

```
dbschema -hd invoices -d pubs_stores_demo
```

Example Output

```

DBSCHEMA Schema Utility          INFORMIX-SQL Version 7.20.UC1
Copyright (C) Informix Software, Inc., 1984-1995
{

Distribution for cath1.invoices.invoice_num

Constructed on 03/10/1995

High Mode, 10.000000 Resolution

--- DISTRIBUTION ---

      (              5)
1: (   16,         7,   11)
2: (   16,         6,   17)
3: (   16,         8,   25)
4: (   16,         8,   38)
5: (   16,         7,   52)
6: (   16,         8,   73)
7: (   16,        12,   95)
8: (   16,        12,  139)
9: (   16,        11,  182)
10: (   10,         5,  200)

--- OVERFLOW ---

1: (    5,          56)
2: (    6,          63)
}

```

Distribution Description

The first part of the **dbschema** output describes which data distributions have been created for the specified table. The name of the table is stated in the following example:

```
Distribution for cath1.invoices.invoice_num
```

The output is for the **invoices** table, which is owned by the user **cath1**. This data distribution describes the column **invoice_num**. If a table has distributions that are built on more than one column, **dbschema** lists the distributions for each column separately.

The date on which the distributions are constructed is listed. In this example, the date is 03/10/1995, which is the date when the UPDATE STATISTICS statement that generated the distributions was executed. You can use this date to tell how outdated your distributions are. Although the system records the date, it does not record the time.

The last line of the description portion of the output describes the mode (medium or high) in which the distributions were created, and the resolution. If you create the distributions with medium mode, the confidence of the sample is also listed. For example, if the UPDATE STATISTICS statement is executed with high mode with a resolution of 10, the last line appears as the following example shows:

```
High Mode, 10.000000 Resolution
```

Distribution Information

The distribution information describes the bins that are created for the distribution, the range of values in the table and in each bin, and the number of distinct values in each bin. Consider the following example:

	(5)
1:	(16,	7,	11)
2:	(16,	6,	17)
3:	(16,	8,	25)
4:	(16,	8,	38)
5:	(16,	7,	52)
6:	(16,	8,	73)
7:	(16,	12,	95)
8:	(16,	12,	139)
9:	(16,	11,	182)
10:	(10,	5,	200)

The first value in the rightmost column is the smallest value in this column. In this example, it is 5.

The column on the left shows the bin number, in this case 1 through 10. The first number in the parentheses shows how many values are in the bin. For this table, 10 percent of the total number of rows (165), is rounded down to 16. The first number is the same for all the bins except for the last. The last row might have a smaller value, indicating that it does not have as many row values. In this example, all the bins contain 16 rows except the last one, which contains 10.

The middle column within the parentheses indicates how many distinct values are contained in this bin. Thus, if there are 11 distinct values for a 16-value bin, it implies that one or more of those values are duplicated at least once.

The right column within the parentheses is the highest value in the bin. The highest value in the last bin is also the highest value in the table. For this example, the highest value in the last bin is 200.

Overflow Information

The last portion of the **dbschema** output shows values that have many duplicates. The number of duplicates of indicated values must be greater than a critical amount that is determined as approximately 25 percent of the resolution times the number of rows. If left in the general distribution data, the duplicates would skew the distribution, so they are moved from the distribution to a separate list, as the following example shows:

```
--- OVERFLOW ---
1: ( 5, 56)
2: ( 6, 63)
```

For this example, the critical amount is $0.25 * 0.10 * 165$, or 4.125. Therefore, any value that is duplicated five or more times is listed in the overflow section. Two values in this distribution are duplicated five or more times in the table: the value 56 is duplicated five times, and the value 63 is duplicated six times.

DB-Access Input from dbschema Output

You can use the **dbschema** utility to get the schema of a database and redirect the **dbschema** output to a file. Later, you can feed this file to DB-Access to re-create the database.

Inserting a Table into a Database Example

The following example copies the CREATE TABLE statements for the customer table into the **dbschema** output file, **tab.sql**:

```
dbschema -d db -t customer > tab.sql
```

Remove the header information about **dbschema** from the output file, **db.sql** and then use DB-Access to re-create the table in another database, as follows:

```
dbaccess db1 tab.sql
```

Re-creating the Schema in Another Database

The following example puts the statements for creating the entire database into the **dbschema** output file:

1. Remove the header information about **dbschema** from the output file, in this case, **db.sql**.
2. Add a CREATE DATABASE statement at the beginning of the output file or use DB-Access to create a new database.
3. Use DB-Access to re-create the schema in a new database:

```
dbschema -d db > db.sql  
dbaccess testdb db.sql
```

4. You can also use the **-ss** option:

```
dbscema -d db -ss > db.sql
```

5. Edit the file **db.sql** and remove the **dbschema** header information.
6. Use DB-Access to re-create the schema in a new database:

```
dbschema testdb db.sql
```

When you use **db.sql** on another database server, confirm that dbspaces exist.

Now you have databases **db** and **testdb**, which differ in name but have the same schema.

Using the LOAD and UNLOAD Statements

In This Chapter	23-3
Syntax of the UNLOAD Statement	23-4
Syntax of the LOAD Statement.	23-5

In This Chapter

This chapter shows the syntax of the SQL UNLOAD and LOAD statements. You can use UNLOAD and LOAD with the following database servers:

- Dynamic Server 2000
- Universal Server
- Extended Parallel Server
- Dynamic Server with AD and XP Options
- OnLine XPS
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE
- OnLine

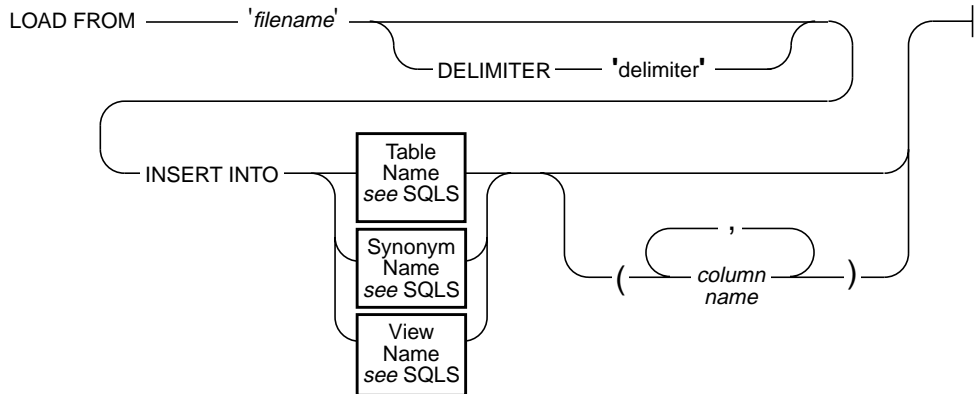
Syntax of the UNLOAD Statement

You can use the UNLOAD statement in DB-Access to unload selected rows from a table into a text file.

UNLOAD TO — 'filename' — DELIMITER — 'delimiter' — SELECT
Statement
see SQLS |

Syntax of the LOAD Statement

You can use the LOAD statement in DB-Access to append rows to an existing table of a database.



The preceding syntax diagrams are only for quick reference. For details about the syntax and use of the UNLOAD and LOAD statements, refer to the *Informix Guide to SQL: Syntax*.

Using the onmode Utility

In This Chapter	24-3
Use of the onmode -b Command for Reversion	24-4
Preparation for Reversion	24-4
Syntax of the onmode -b Command	24-5

In This Chapter

This chapter describes the **-b** option of the **onmode** utility and how to use it. You can use **onmode -b** with the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server
- SE
- OnLine

You can use the **-b** option of the **onmode** utility for reversion from an upgraded database server to the earlier database server. The **onmode** utility modifies the data in an Informix database so that the earlier version of the database server can access it. For information about the other **onmode** options, refer to your *Administrator's Guide*.

UNIX

WIN NT

Use of the `onmode -b` Command for Reversion

When you upgrade a database server, several modifications make the format of the databases incompatible with the older version. The **`onmode -b`** command restores the databases to a format that is compatible with the earlier version. You must revert the databases before users can access the data with the earlier database server version. The utility does not revert changes made to the layout of the data that do not affect compatibility.

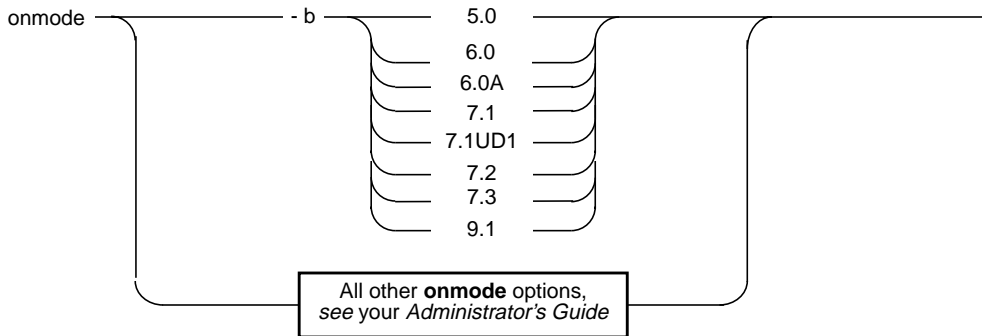
You must be user **root** or user **informix** to execute **`onmode`**. ♦

You must be a member of the **Informix-Admin** group to execute **`onmode`**. ♦

Preparation for Reversion

Before you use the **`-b`** option, notify users that you are going to bring the database server off-line. The reversion utility forcibly removes all users and shuts down the database server. The **`-b`** option includes an implicit **`-yuk`**. Make sure that the **INFORMIXSERVER** environment variable is set to the correct database server.

Syntax of the onmode -b Command



Element	Purpose	Key Considerations
-b 5.0	Change the database to the Version 5.0 format.	Additional Information: Refer to “Reverting to OnLine 5.0 or 4.1” on page 12-39.
-b 6.0	Change the database to the Version 6.0 format.	Additional Information: Refer to “Reverting to an Earlier Version of OnLine Dynamic Server” on page 11-36.
-b 6.0A	Change the database to the Version 6.0 ALS format.	Additional Information: Refer to “Reverting to Version 6.x ALS” on page 19-6.
-b 7.1	Change the database to the Version 7.10.UD1 format, which is compatible with all 7.10.UDx formats.	Additional Information: Refer to “Reverting to an NLS Database Server” on page 19-4 and “Reverting to OnLine Dynamic Server 7.10.UDx from OnLine Dynamic Server 7.10.UD1 Through 7.14” on page 11-53.
-b 7.1UD1	Change the database to the Version 7.1UD1 format, which is compatible with 7.11, 7.12, 7.13, and 7.14 formats.	Additional Information: Refer to “Reverting to an Earlier Version of OnLine Dynamic Server” on page 11-36.

(1 of 2)

Element	Purpose	Key Considerations
-b 7.2	Change the database to the Version 7.2x format.	Additional Information: Refer to “Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000” on page 4-26, “Reverting to OnLine Dynamic Server 7.2x or 7.1x from Universal Server” on page 6-20, and “Reverting to an Earlier Version of OnLine Dynamic Server” on page 11-36.
-b 7.3	Change the database to the Version 7.3x format.	Additional Information: Refer to “Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000” on page 4-26.
-b 9.1	Change the database to the Version 9.14 format.	Additional Information: Refer to “Reverting to Universal Server 9.14, Dynamic Server 7.3x, or OnLine Dynamic Server 7.2x from Dynamic Server 2000” on page 4-26.

(2 of 2)



Tip: If you type *onmode -b -*, the available options appear.



Important: You cannot use redirection to a file when you execute *onmode -b 7.2* because this command does not function with redirection.

Using the onunload and onload Utilities

In This Chapter	25-3
How onunload and onload Work	25-4
Syntax of the onunload Command	25-5
Syntax of the onload Command	25-10
Constraints That Affect onload and onunload	25-13
Steps for Using onunload and onload	25-16



XPS 8.3

AD/XP

XPS 8.11

In This Chapter

This chapter describes the **onunload** and **onload** utilities and how to use them. You can use **onunload** and **onload** with the following database servers:

- Dynamic Server 2000
- Universal Server
- Dynamic Server 7.3
- Dynamic Server, Workgroup and Developer Editions
- Dynamic Server, Linux Edition
- OnLine Dynamic Server
- OnLine Workgroup Server

***Important:** You can use **onunload** and **onload** with Dynamic Server 2000 or Universal Server if the databases contain only legacy data types and not any extended data types.*

To load and unload data in Extended Parallel Server, use the **onxfer** utility, which [Chapter 26](#) describes. ♦

To load and unload data in Dynamic Server with AD and XP Options or OnLine XPS, use external tables formatted in the Informix internal data representation format. You can load and unload files with the default delimiter (|) format. For more information on external tables, see your *Administrator's Guide* or the *Administrator's Reference*. For syntax, see the *Informix Guide to SQL: Syntax*. ♦

The **onunload** and **onload** utilities unload and load databases and tables.

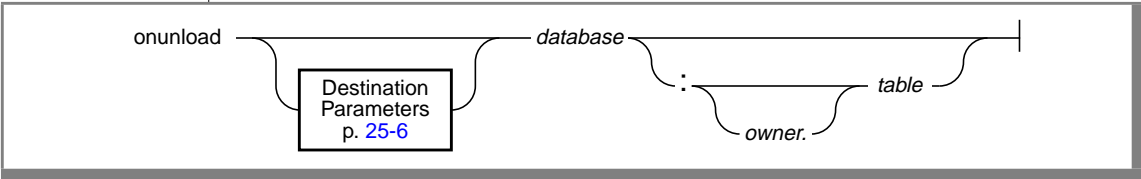
How onunload and onload Work

The **onunload** utility unloads data from a database. The **onunload** utility writes a database or table into a file on tape or disk. The **onunload** utility unloads the data in binary form in disk-page units, making this utility more efficient than **dbexport**. You can use the **onunload** utility to move data between computers.

The **onload** utility loads data that was created with the **onunload** command into the database server. The **onload** utility creates a database or table in a specified dbspace. Then **onload** loads it with data from an input tape or disk file that the **onunload** utility creates.

During the load, you can move simple large objects that are stored in a blob space to another blob space.

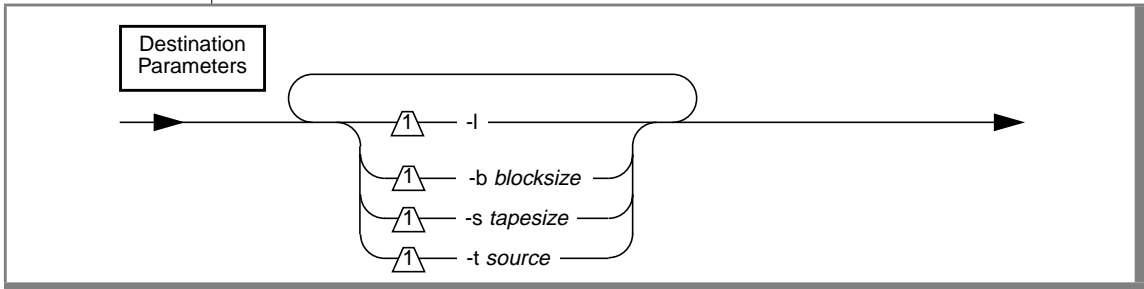
Syntax of the onunload Command



Element	Purpose	Key Considerations
<i>database</i>	Specifies the name of a database.	Additional Information: The database name cannot include a database server name (<i>database@dbservername</i>). References: Syntax must conform to the Identifier segment; see the <i>Informix Guide to SQL: Syntax</i> .
<i>owner.</i>	Specifies the owner of the table.	Additional Information: The owner name must not include illegal characters. References: For pathname syntax, see your operating-system documentation.
<i>table</i>	Specifies the name of the table.	Restriction: The table must exist. References: Syntax must conform to the Table Name segment; see the <i>Informix Guide to SQL: Syntax</i> .

If you do not specify any destination parameter options, **onunload** uses the device that TAPEDEV SPECIFIES. The block size and tape size are the values specified as TAPEBLK and TAPESIZE, respectively. (For information about TAPEDEV, TAPEBLK, and TAPESIZE, refer to your *Administrator's Guide*.)

Destination Parameters



Element	Purpose	Key Considerations
-b <i>blocksizes</i>	Specifies in kilobytes the block size of the tape device.	Restrictions: The <i>blocksizes</i> must be an integer. Additional Information: This option overrides the default value in TAPEBLK or LTAPEBLK.
-l	Directs onunload to read the values for tape device, block size, and tape size from LTAPEDEV, LTAPEBLK, and LTAPESIZE, respectively.	None.
-s <i>tapesize</i>	Specifies in kilobytes the amount of data that can be stored on the tape.	Restrictions: The <i>tapesize</i> must be an integer. Additional Information: This option overrides the default value in TAPESIZE or LTAPESIZE.
-t <i>source</i>	Specifies the pathname of the file on disk or of the tape device where the input tape is mounted.	Additional Information: This option overrides the tape device specified by TAPEDEV or LTAPEDEV. It must be a legal pathname.

IDS 2000

IUS

Constraints That Affect *onunload*

The **onunload** utility can unload data more quickly than either **dbexport** or the UNLOAD statement because it copies the data in binary and in page-sized units. However, this feature places the following constraints on its use:

- You must load the data on the **onunload** tape into a database or table that your database server (excluding SE) manages.
- You can use **onunload** and **onload** with Dynamic Server 2000 or Universal Server if the databases contain legacy data types. ♦
- You must load the tape that **onunload** writes onto a computer with the same page size and the same representation of numeric data as the original computer.
- You must read the file that **onunload** creates with the **onload** utility of the same version of your database server. You cannot use **onunload** and **onload** to move data from one version to another.
- When you unload a complete database, you cannot modify the ownership of database objects (such as tables, indexes, and views) until after you finish reloading the database.
- When you unload and load a table, **onunload** does not preserve access privilege, synonyms, views, constraints, triggers, or default values that were associated with the original tables. Before you run **onunload**, use the **dbschema** utility to obtain a listing of the access privilege, synonyms, views, constraints, triggers, or default values. After you finish loading the table, use **dbschema** to re-create the specific information for the table.

Database or Table Unloading

To unload a database, you must have DBA privileges for the database or be user **informix**. To unload a table, you must either own the table, have DBA privileges for the database in which the table resides, or be user **informix**. (User **root** does not have special privileges with respect to **onunload** and **onload**.)

Unloading a Database

If you unload a database, all the tables in the database, including the system catalog tables, are unloaded. All the triggers, SPL routines, defaults, constraints, and synonyms for all the tables in the database are also unloaded.

Unloading a Table

If you unload a table, **onunload** unloads the table data and information from the following system catalog tables:

- **systables**
- **syscolumns**
- **sysindexes**
- **sysblobs**

When you unload a table, **onunload** does not unload information about constraints, triggers, or default values that are associated with a table. In addition, access privileges that are defined for the table and synonyms or views that are associated with the table are not unloaded.

Logging Mode

The **onunload** utility does not preserve the logging mode of a database. After you load the database with **onload**, you can make a database ANSI compliant or add logging. For information about logging modes, refer to the *Informix Guide to SQL: Syntax*.

During the load, you can move simple large objects that are stored in a blobspace to another blobspace.

If you do not specify any source-parameter options, **onload** uses the device that is specified as TAPEDEV. The block size and tape size are the values that are specified as TAPEBLK and TAPESIZE, respectively. (For more information about TAPEDEV, TAPEBLK, and TAPESIZE, refer to your *Administrator's Guide*.)

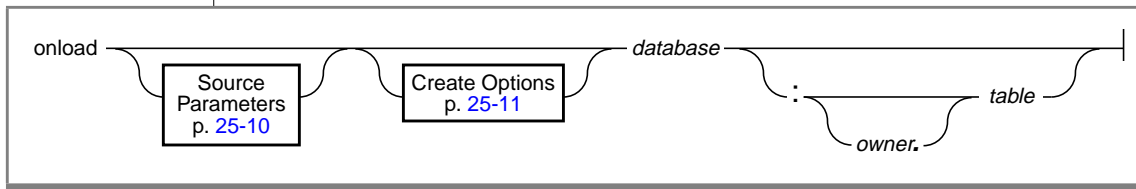
If you do not specify creation options, **onload** stores the database or table in the root dbspace.

Locking During Unload Operation

During the unload operation, the database or table is locked in shared mode. An error is returned if **onunload** cannot obtain a shared lock.

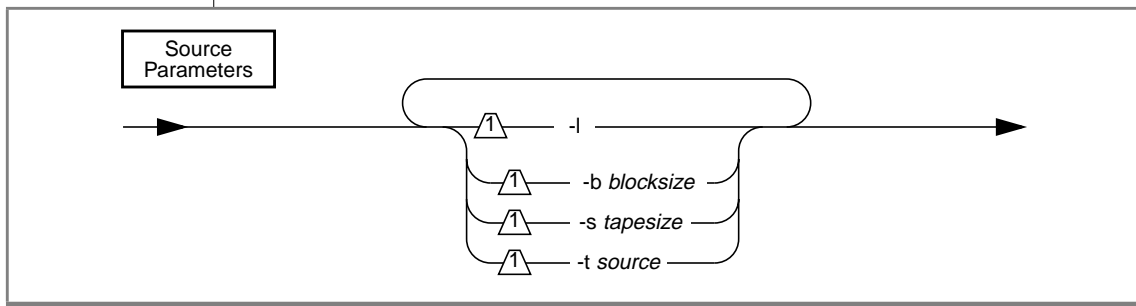
The **onload** utility creates a database or table in a specified dbspace (excluding SE). Then **onload** loads it with data from an input tape or disk file that the **onunload** utility creates.

Syntax of the onload Command



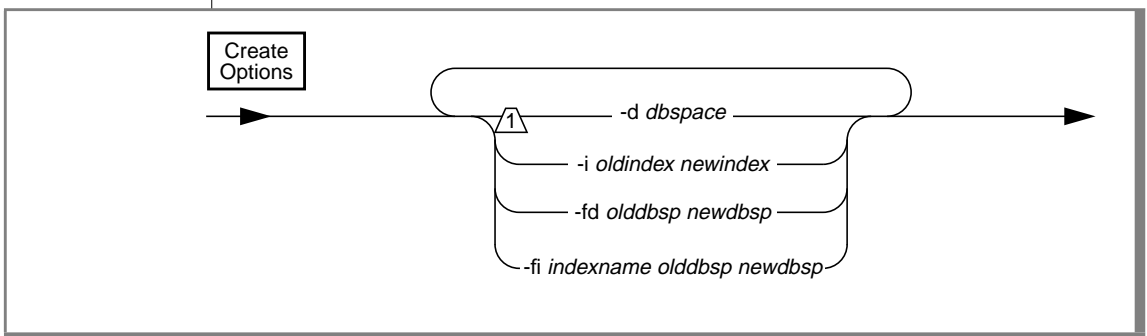
Element	Purpose	Key Considerations
<code>database</code>	Specifies the name of the database.	Restriction: The database name cannot include a database server name (<code>database@dbservername</code>). References: Syntax must conform to the Identifier segment; see the <i>Informix Guide to SQL: Syntax</i> .
<code>owner.</code>	Specifies the owner of the table.	Restriction: The owner name must not include illegal characters. References: For pathname syntax, refer to your operating-system documentation.
<code>table</code>	Specifies the name of the table.	Restriction: The table must not exist. References: Syntax must conform to the Table Name segment; see the <i>Informix Guide to SQL: Syntax</i> .

Source Parameters



Element	Purpose	Key Considerations
-b <i>blocksize</i>	Specifies in kilobytes the block size of the tape device.	Restrictions: Unsigned integer. Must specify the block size of the tape device. Additional Information: This option overrides the default value in TAPEBLK or LTAPEBLK.
-l	Directs onload to read the values for tape device, block size, and tape size from the configuration parameters LTAPEDEV, LTAPEBLK, and LTAPESIZE, respectively.	Additional Information: If you specify -l and then -b , -s , or -t , the value that you specify overrides the value in the configuration file.
-s <i>tapesize</i>	Specifies in kilobytes the amount of data that the database server can store on the tape.	Restrictions: Unsigned integer. Must specify the amount of data that the database server can store on the tape. Additional Information: This option overrides the default value in TAPESIZE or LTAPESIZE.
-t <i>source</i>	Specifies the pathname of the file on disk or of the tape device where the input tape is mounted.	Restriction: Must be a legal pathname. Additional Information: This option overrides the tape device that TAPEDEV or LTAPEDEV specifies. References: For pathname syntax, see your operating-system documentation.

Create Options



If you do not specify any create options, the **onload** utility stores the database or table in the root dbspace.

You can use the **-d**, **-i**, **-fd**, and **-fi** options in any order and as often as necessary as long as you use unique pairs.

Constraints That Affect onload

The **onload** utility performs faster than the **dbimport**, **dbload**, or **LOAD** methods. In exchange for this higher performance, **onload** has the following constraints:

- The **onload** utility can only create a new database or table; you must drop or rename an existing database or table of the same name before you run **onload**. The **onload** utility prompts you to rename blobspaces during execution, if desired.
- The **onload** utility places a shared lock on each of the tables in the database during the load. While you cannot update a table row with the lock in place, the database is available for queries.
- When you load a complete database, the user who executes **onload** becomes the owner of the database.
- The **onload** utility creates a database without logging; you must initiate logging after **onload** loads the database.
- When you use **onload** to load a table into a logged database, you must turn off logging for the database during the operation.

GLS

NLS

Constraints That Affect onload and onunload

You can use **onunload** and **onload** to move data between databases if the NLS and GLS locales are identical. For example, if user A has a French locale NLS table on server A and tries to load data into a German locale GLS table on server B, **onload** and **onunload** report errors. However, if both the NLS and GLS tables were created with the same French locale, **onload** and **onunload** would work. ♦

The tape that **onload** reads contains binary data that is stored in disk-page-sized units. For this reason, the computers where the original database resides (where you use **onunload**) and where the target database will reside (where you use **onload**) must have the following characteristics:

- The same page size
- The same representation of numeric data
- The same byte alignment for structures and unions

If the page sizes are different, **onload** fails. If the alignment or numeric data types on the two computers are different (for example, with the most significant byte last instead of first, or different float-type representations), the contents of the data page could be misinterpreted.

Restrictions That Affect onload and onunload

The **onload** and **onunload** utilities have the following restrictions:

- The original database and the target database must be from the same version of the database server.
- You cannot use **onload** and **onunload** to move data between non-GLS and GLS locales. ♦
- Do not use **onload** and **onunload** to move data between two Universal Server databases if they contain extended data types. Use the HPL instead to move Dynamic Server 2000 or Universal Server data. However, you can use **onload** and **onunload** with this data if the databases contain only legacy data types. ♦
- SE does not support **onload** and **onunload**. ♦

GLS

IDS 2000

IUS

SE



Important: You cannot use the **onload** and **onunload** utilities to move data from one version to another. You also cannot use these utilities to move data between different types of servers.

Logging During Loading

The **onload** utility performs all its loading within a transaction. This feature allows the changes to be rolled back if an error occurs.

When you use **onload** to create tables from an **onunload** input tape, **onload** can only load information into a database without logging. Thus, before you load a table into an existing, logged database, end logging for the database. You also might want to consider loading during off-peak hours. Otherwise, you might fill the logical-log files or consume excessive shared-memory resources. After you load the table, create a level-0 dbspace backup before you resume database logging.

When you use **onload** to create databases from an **onunload** input tape, the databases that result are not ANSI compliant and do not use transaction logging. You can make a database ANSI compliant or add logging after you load the database. (For more information about logging, refer to the *Informix Guide to SQL: Reference*.)

Movement of Simple Large Objects to a Blobspace

If you load a table that contains simple large objects stored in a blobspace, **onload** asks you if you want to move them to another blobspace. If you respond *yes*, **onload** displays the blobspace name where the simple large objects were stored when the tape was created. It then asks you to enter the name of the blobspace where you want the simple large objects stored. If you enter a valid blobspace name, **onload** moves all simple-large-object columns in the table to the new blobspace. Otherwise, **onload** prompts you again for a valid blobspace name.

Ownership and Privileges

When you load a new database, the user who runs **onload** becomes the owner. Ownership within the database (tables, views, and indexes) remains the same as when the database was unloaded to tape with **onunload**.

To load a table, you must have the Resource privilege on the database. When **onload** loads a new table, the user who runs **onload** becomes the owner unless you specify an owner in the table name. (You need the DBA privilege for the database to specify an owner in the table name.)

The **onunload** utility does not preserve synonyms or access privileges. To obtain a listing of defined synonyms or access privileges, use the **dbschema** utility, which [Chapter 22, “Using the dbschema Utility,”](#) describes, before you run **onunload**.

Exclusive Locking During Load Operation

During the load operation, **onload** places an exclusive lock on the new database or table. Loading proceeds as a single transaction, and **onload** drops the new database or table if an error or system failure occurs.

Steps for Using *onunload* and *onload*

This section describes the procedure for using **onunload** and **onload** to move a database. You can use these commands to move either a complete database or a table from one computer to another. The syntax and description of the **onunload** utility starts on [page 25-5](#). The syntax and description of the **onload** utility starts on [page 25-10](#).

To move a database from one computer to another

1. Make sure that the page size, numeric representations, and byte alignment on structures and unions are the same on both computers.
(The page size is 2 kilobytes on certain UNIX systems and 4 kilobytes on Windows NT.) The page size is an Informix characteristic. For information about page size, refer to your *Administrator's Guide*. The numeric representation and the byte alignment are characteristics of your operating system. For information about numeric representation and byte alignment, refer to the manuals for your operating systems.
2. Decide where to store the unloaded data.
 - On disk Create an empty file for **onunload** to hold the data. Make sure that you have write permission for the file.
 - On tape Use the tape device and characteristics specified in the ONCONFIG configuration file by either TAPEDEV or LTAPEDEV or specify another tape device. Make sure that the tape device that you specify is available for **onunload**.
3. Run the **oncheck** utility to make sure that your database is consistent.
For information about **oncheck**, refer to your *Administrator's Guide*.
4. If you want to save the triggers, access privileges, SPL routines, defaults, constraints, and synonyms for the tables in the database, run the **dbschema** utility.
5. Run the **onunload** utility.
6. If necessary, transfer the storage medium (tape or disk) to the new computer.

Using the onxfer Utility

In This Chapter	26-3
Environment Variables for onxfer	26-4
Syntax of the onxfer Command	26-5
Configuration File for onxfer	26-7
Guidelines and Restrictions for Using onxfer	26-17

In This Chapter

This chapter describes the **onxfer** utility and how to use it. You can use **onxfer** with the following database servers:

- Extended Parallel Server (move data to only)
- Dynamic Server with AD and XP Options (move data to only)
- Dynamic Server 7.3 (move data from only)
- Dynamic Server, Workgroup and Developer Editions (move data from only)
- Dynamic Server, Linux Edition (move data from only)
- OnLine Dynamic Server (move data from only)

This chapter describes how to use the **onxfer** utility for moving data to Extended Parallel Server or Dynamic Server with AD and XP Options from Dynamic Server 7.3x or OnLine Dynamic Server 7.24 on UNIX. The **onxfer** utility moves a database or tables to the target database server from the source database server.

Environment Variables for onxfer

The **onxfer** utility uses the following environment variables:

- **SOURCE_REMOTE_SHELL**
- **XFER_CONFIG**

SOURCE_REMOTE_SHELL Environment Variable

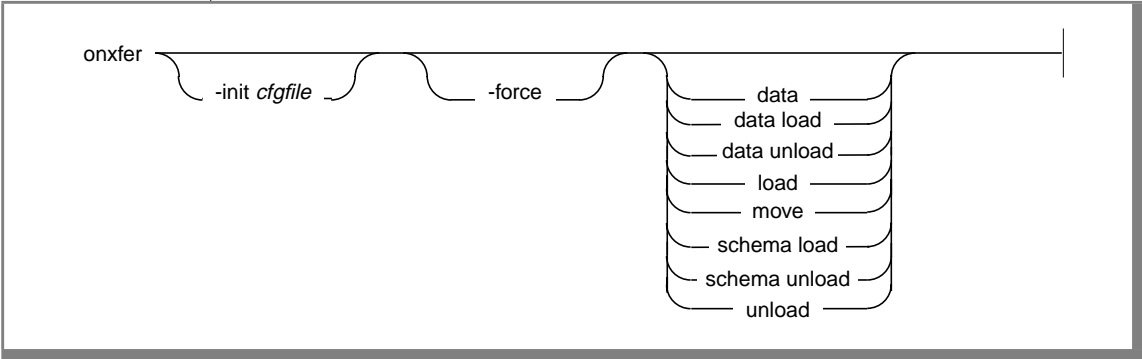
When the source and target database servers are installed on different computers, **onxfer** might use a remote shell command on the source computer to perform certain operations. However, the name of the remote shell command might be different on different computers.

If the **onxfer** utility is not able to find the correct path name for the remote shell command, **onxfer** uses **rsh** as the default remote shell command. You can set the **SOURCE_REMOTE_SHELL** environment variable to specify the remote shell command that the operating system supports on the source computer. If this environment variable is set, **onxfer** uses its value as the remote shell command.

XFER_CONFIG Environment Variable

The default name of the **onxfer** configuration file is **xfer_config** in the **\$INFORMIXDIR/etc** directory. You can specify a different directory in the **XFER_CONFIG** environment variable.

Syntax of the onxfer Command



Element	Purpose	Key Considerations
data	Moves the data transparently.	The onxfer utility moves the tables iteratively.
data load	Loads data from disk.	
data unload	Unloads all the specified files to disk.	You can use this option only when the DEVICE configuration parameter is set to DISK. See “DEVICE Configuration Parameter” on page 7.
-force	<p>Before unloading data or schema onto disks, onxfer checks to see if the directories already contain previously unloaded schema or data. If they do, the default behavior is to not perform the unload operation and to ask the user to validate the unload operation by cleaning up the directories first.</p> <p>You can use the -force option to override the default behavior and force the unload operation even if the directories contain unloaded schema or data. If you use this option, onxfer deletes any previously unloaded schema and data from the directories that you specify in the DISK option and performs a fresh unload operation.</p>	

Syntax of the onxfer Command

Element	Purpose	Key Considerations
-init <i>cfgfile</i>	Specifies a configuration file that contains parameters for onxfer .	The default configuration file is xfer_config . You can edit this file to change onxfer parameters or make your own copy of the file and specify it in the onxfer command. The XFER_CONFIG environment variable specifies the location of the xfer_config file. The default location is \$INFORMIXDIR/etc .
load	Loads both the schema and the data from disk.	
move	Moves the data and the schema transparently.	The onxfer utility moves the tables iteratively. You need to edit the schema.
schema load	Loads the schema into the target database server.	The schema load option uses the same configuration file that schema unload used.
schema unload	Unloads the source schema to a file.	
unload	Unloads both the schema and the data from disk.	

(2 of 2)

Configuration File for onxfer

You can specify a configuration file in the **onxfer** command. The configuration file includes the configuration parameters to use for data movement. If you do not specify a configuration file, **onxfer** uses the default configuration file, **xfer_config**.

The default location of the **xfer_config** file is **\$INFORMIXDIR/etc**. You can specify a different location in the **XFER_CONFIG** environment variable.

An **onxfer** configuration file can include the following configuration parameters:

- DEVICE
- FORMAT
- MSGPATH
- NSTREAMS
- REJECT_DIR
- SCHEMA_DIR
- SOURCE and TARGET
- SOURCE_DIR
- TARGET_DIR

DEVICE Configuration Parameter

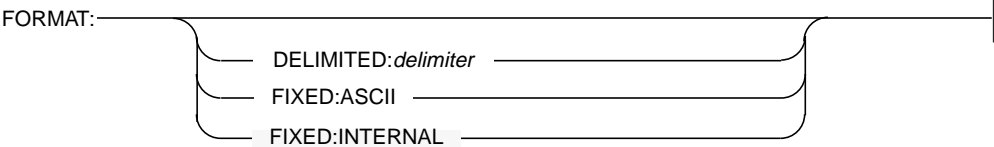
The **DEVICE** configuration parameter specifies a destination device and can specify destination directories. This configuration parameter is optional.

DEVICE — DISK:directory_list
DIRECT

Element	Purpose	Key Considerations
DISK: <i>directory_list</i>	Specifies the directories for unloading data.	<p>The number of directories should equal the value of NSTREAMS, which specifies the number of parallel data streams. A directory name cannot exceed 18 characters. See “NSTREAMS Configuration Parameter” on page 26-11.</p> <p>The default <i>directory_list</i> is a number of subdirectories under the SCHEMA_DIR directory in the form database.data.suffix, in which database is the name of the source database and suffix ranges from 1 to NSTREAMS. See “SCHEMA_DIR Configuration Parameter” on page 26-13.</p> <p>The directories need enough disk space to hold all the unloaded data.</p> <p>You must specify DISK for all onxfer command-line options except -move and except when you use the -data option without the load or unload suboption.</p>
DIRECT	Specifies direct transfer of table data using the network and pipes.	<p>For schema transfer, onxfer uses the disk directory that the SCHEMA_DIR parameter specifies. See “SCHEMA_DIR Configuration Parameter” on page 26-13.</p> <p>You must specify DIRECT for the -move command-line option and when you use the -data option without the load or unload suboption. See “Syntax of the onxfer Command” on page 26-5.</p>

FORMAT Configuration Parameter

The FORMAT configuration parameter specifies the format of the data during transfer. This configuration parameter is optional.



Element	Purpose	Key Considerations
DELIMITED: <i>delimiter</i>	Suggests the use of ASCII-delimited format to transfer data. The value of <i>delimiter</i> is a field delimiter.	The row delimiter is always the new-line character.
FIXED:ASCII	Suggests the use of FIXED format ASCII representation to transfer data.	
FIXED:INTERNAL	Indicates that the loaders have to use Informix internal format to transfer data.	FIXED:INTERNAL, the default FORMAT option, provides the fastest transfer mechanism. This format does not work well for certain data types, such as DATETIME, INTERVAL, and MONEY.

MSGPATH Configuration Parameter

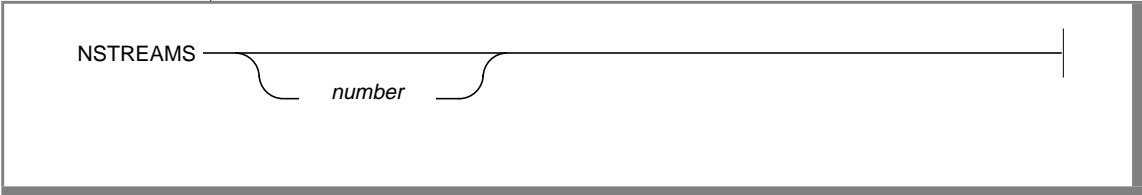
The MSGPATH configuration parameter specifies the location of the **onxfer** message log. This configuration parameter is optional. This onxfer configuration parameter is different from the onconfig MSGPATH configuration parameter.



Element	Purpose	Key Considerations
<i>pathname</i>	Specifies the name of the message log where the onxfer utility can log errors and post progress messages.	The default message log file is database.msg in the SCHEMA_DIR directory, in which database is the name of the source database. See “SCHEMA_DIR Configuration Parameter” on page 13.

NSTREAMS Configuration Parameter

The NSTREAMS configuration parameter specifies the number of parallel streams for moving data, one stream for each coserver. This configuration parameter is optional.



Element	Purpose	Key Considerations
<i>number</i>	Suggests the number of parallel streams to use for unloading or loading a table.	The default value of NSTREAMS is the number of coservers in Extended Parallel Server. The onxfer utility feeds each stream of data to one coserver.

REJECT_DIR Configuration Parameter

The REJECT_DIR configuration parameter specifies a location for reject files, which contain incompatible SQL statements after data unloading or loading. This configuration parameter is optional.



Element	Purpose	Key Considerations
<i>pathname</i>	Specifies the directory in which onxfer can create reject files for unload and load operations.	<p>The default directory for reject files is SCHEMA_DIR. The REJECT_DIR <i>pathname</i> cannot exceed 18 characters. See “SCHEMA_DIR Configuration Parameter” on page 26-13.</p> <p>Reject file names are in the form database.rej.suffix, in which database is the name of the source database and suffix is a number from 1 to the value of NSTREAMS. See “NSTREAMS Configuration Parameter” on page 26-11.</p>

SCHEMA_DIR Configuration Parameter

The SCHEMA_DIR configuration parameter specifies a destination directory for the schema file. The path name for the destination directory is the path from the computer on which the target database server is installed, where **onxfer** is running. This configuration parameter is required.



Element	Purpose	Key Considerations
path_name	Specifies the directory into which onxfer is to unload the schema file.	<p>The name of an unloaded schema file is database.sql, in which database is the name of the source database.</p> <p>You need to change the contents of the schema file to suit your database server requirements.</p>

SOURCE and TARGET Configuration Parameters

The SOURCE configuration parameter can specify individual tables to transfer as well as the source database and database server. The TARGET configuration parameter can specify the target database and database server. The **onxfer** configuration file can contain any number of SOURCE and TARGET configuration parameters.

The diagram illustrates the structure of SOURCE and TARGET configuration parameters in a configuration file. It uses brackets to group specific parts of the syntax for clarity.

SOURCE Configuration:

- `SOURCE` is followed by a bracketed group containing `database @` and `server_name`.
- Below this is a large empty bracketed box, likely for additional source-specific options.
- Then, a bracketed group contains `TABLE LIST:table_list` and `%END-TABLE`.
- Below that, another bracketed group contains `TABLE QUERY tabname operator pattern`.
- Next, a bracketed group contains `TARGET_DIR $INFORMIXDIR` and `SOURCE_DIR $INFORMIXDIR`.
- Finally, a bracketed group contains `%END-SOURCE`.

TARGET Configuration:

- `TARGET` is followed by a bracketed group containing `database @` and `server_name`.

Element	Purpose	Key Considerations
<i>database @</i>	Names the source or target database.	<p>If you omit the source database name, onxfer transfers the entire instance of the source database server to the target. In this case, the sysmaster database must exist on the source database server.</p> <p>The target database name can be the same as or different from the source name.</p> <p>The onxfer utility transfers the schema for the entire database and data from selected tables. If you do not specify any tables in the TABLE LIST or TABLE QUERY option, onxfer transfers all the data from the source database server.</p>
<i>server_name</i>	Names the source or target database server.	
TABLE LIST <i>table_list</i>	Lists the database tables from which to transfer data.	If you do not specify any tables, onxfer transfers all the data from the source database server.
TABLE QUERY <i>tablename operator pattern</i>	Specifies part of a WHERE clause for table selection.	If you do not specify any tables, onxfer transfers all the data from the source database server.

SOURCE_DIR Configuration Parameter

The SOURCE_DIR configuration parameter specifies the source database server. The path name for the source database server is the path from the computer on which the source database server is installed. This configuration parameter is required.

SOURCE_DIR — \$INFORMIXDIR

TARGET_DIR Configuration Parameter

The TARGET_DIR configuration parameter specifies the target database and database server. The path name for the target database server is the path from the computer on which the target database server is installed, where **onxfer** is running. This configuration parameter is required.

TARGET_DIR — \$INFORMIXDIR

Guidelines and Restrictions for Using onxfer

The following guidelines and restrictions provide information about using **onxfer**:

- You must execute the **onxfer** command from the computer on which the target database server is installed. When invoked, the **\$INFORMIXDIR** environment variable should point to the installed target database server distribution.
- Currently, **onxfer** supports the **unload** and **data unload** command-line options when you execute it from the computer on which the source database server is installed, the same computer on which the target database server is installed.
- For the **load** and **data load** command-line options, the directory path names that you specify in the DISK device option must be the path names as seen on the computer(s) on which the target database server instance is installed. The directories, however can be NFS mounted from other computers.
- When you use the **move** command-line option, **onxfer** optimizes the speed of data transfer by first creating all the tables with the RAW type on the target database server instance, by executing the **<database>_pre.sql** file from the SCHEMA_DIR destination directory.

The **onxfer** utility then transfers data in iterative mode, one table at a time, from the source database to the target database using the EXPRESS mode of loading. Finally, **onxfer** creates the rest of the schema by executing the **<database>_post.sql** file from the SCHEMA_DIR destination directory.

When, however, you specify the **data** or **data load** command-line option, **onxfer** cannot use the EXPRESS mode of loading because the tables are typically not of the RAW type and there might be pre-existing constructs like indexes, check constraints, and triggers defined on the tables. **Onxfer** uses DELUXE mode to load data in such cases.

- The DELUXE mode of loading can be considerably slower than the EXPRESS mode. Also, in DELUXE mode, the insertion of each record into the table and corresponding index updates are logically logged. Each table load is treated as a unit of transaction.

This requires creation of logical log space large enough to hold log records for the single largest table load before an attempt to load data with the **data** or **data load** option.

- The NSTREAMS parameter, when specified, applies to all the tables in the TABLE section.
- The NSTREAMS parameter should not be set to a value greater than ((number of CPU VPs per co-server) * (number of co-servers)) as configured on the target database server instance.
- Do not use the co-server extension while specifying the TARGET database server name (for example, do not use **my_db@my_adxp.2**). Use the prefix without the extension instead (for example, use **my_db@my_adxp**).
- Do not modify or delete files under the directories that **onxfer** uses to load or unload data.
- Do not modify or delete any file under the SCHEMA_DIR destination directory, except **<database>_pre.sql** and **<database>_pre.sql**, which you can edit as needed.
- Before completing execution, **onxfer** does the necessary cleanup whether the operation was successful or not. In case of certain fatal errors, **onxfer** might not be able to do so and can flag an error message for users to perform cleanup manually.

In such cases, you might need to clean up the following objects:

- Source database server object
The **onxfer** database, typically set to **xferdb_####**, in which **####** is the process ID of the **onxfer** program
- Target database server objects
The **ifmx_xfer_status** database, used to store results of pipe-based data transfers
External table definitions in the target database that correspond to the tables listed under TABLE section
- Disk objects
Files under the SCHEMA_DIR destination directory
Files under directories specified in the DISK option, if any

Database Server Environment Variables

Various *environment variables* affect the functionality of your Informix products. You can set environment variables that identify your terminal, specify the location of your software, and define other parameters. The environment variables discussed in this chapter are listed alphabetically beginning on [page A-3](#).

Some environment variables are required, and others are optional. For example, you must set or accept the default setting for certain UNIX or Windows NT environment variables.

This chapter describes how to use the environment variables that apply to Informix database servers and shows how to set them.

UNIX

WIN NT

GLS

NLS

Types of Environment Variables

The environment variables discussed in this chapter fall into the following categories:

- **Informix environment variables**
Set these standard environment variables when you want to work with Informix products. Each product manual specifies the environment variables that you must set to use that product.
- **UNIX environment variables**
Informix products rely on the correct setting of certain standard UNIX system environment variables. The **PATH** and **TERM** environment variables must always be set. You might also have to set the **TERMCAP** or **TERMINFO** environment variable to use some products effectively. ♦
- **Windows NT environment variables**
Informix products rely on the correct setting of certain standard Windows NT system environment variables. The **PATH** environment variable must always be set. ♦
- **GLS environment variables**
The GLS environment variables that enable you to work in a nondefault locale are described in the *Informix Guide to GLS Functionality*. However, these variables are also included in the list of environment variables [on page A-3](#). ♦
- **NLS environment variables (pre-7.2 products only)**
You must set some or all of these X/Open standard environment variables to benefit from NLS. These environment variables might cause your product to behave differently than when their standard Informix counterparts are set. Refer to [“Native Language Support” on page 17-9](#). ♦

List of Environment Variables

Figure A-1 contains an alphabetical list of the environment variables that you can set for an Informix database server. For instructions and detailed descriptions on setting these environment variables, see the *Informix Guide to SQL: Reference* for your database server. The following table uses these abbreviations for the database server names and Informix manuals:

- IDS 2000 (Informix Dynamic Server 2000)
- XPS 8.3 (Informix Extended Parallel Server)
- IDS 7.3 (Informix Dynamic Server 7.3)
- IUS (INFORMIX-Universal Server)
- OWS (Informix Dynamic Server, Workgroup and Developer Editions, or INFORMIX-OnLine Workgroup Server)
- WE (Informix Dynamic Server, Workgroup Edition)
- AD/XP (Informix Dynamic Server with Advanced Decision Support and Extended Parallel Options)
- XPS 8.11 (INFORMIX-OnLine XPS)
- ODS (INFORMIX-OnLine Dynamic Server)
- SE (INFORMIX-SE)
- GLS (*Informix Guide to GLS Functionality*)
- REF (*Informix Guide to SQL: Reference*)
- ESQL/C (*Informix ESQL/C Programmer's Manual*)

GLS

The GLS environment variables are discussed in the *Informix Guide to GLS Functionality*. ♦

Figure A-1
Environment Variables for Informix Database Servers

Environment Variable	Restrictions	Reference
ARC_DEFAULT	IDS 2000, IDS 7.3, IUS, ODS only (all platforms)	REF
ARC_KEYPAD	IDS 2000, IDS 7.3, IUS, ODS only (all platforms)	REF
CC8BITLEVEL	ESQL/C only	REF
CLIENT_LOCALE	GLS only	GLS
COLLCHAR	NLS only	REF
DBANSIWARN		REF
DBAPICODE	NLS only	REF
DBBLOBBUF	IDS 2000, IDS 7.3, IUS, ODS only (all platforms)	REF
DBCENTURY	SQL APIs only	REF
DBDATE		REF, GLS
DBDELIMITER		REF
DBEDIT		REF
DBFLTMASK	DB-Access only	REF
DBLANG		REF, GLS
DBMONEY		REF, GLS
DBNLS	NLS only	REF
DBONPLOAD	The High-Performance Loader only	REF
DBPATH		REF
DBPRINT	UNIX only	REF
DBREMOTECMD	IDS 2000, IDS 7.3, IUS, ODS, OWS only (UNIX)	REF

(1 of 4)

Environment Variable	Restrictions	Reference
DBSPACETEMP	IDS 2000, IDS 7.3, IUS, ODS, OWS only	REF
DBTEMP	SE only	REF
DBTIME	SQL APIs only	REF
DBUPSPACE		REF
DB_LOCALE	GLS only	GLS
DELIMIDENT		REF
ENVIGNORE		REF
ESQLMF	GLS only	GLS
FET_BUF_SIZE	SQL APIs and DB-Access only	REF
FX_DIRECTIVES		REF
GLS8BITFSYS	SE only, GLS only	GLS
GL_DATE	GLS only	GLS
GL_DATETIME	GLS only	GLS
IFX_AUTOFREE	Informix ESQL/C Version 7.23 or later client applications	ESQL/C
IFX_DIRECTIVES	IDS 2000 only	REF
IFX_LONGID	IDS 2000 only	REF
INFORMIXC	ESQL/C only	REF
INFORMIXCONRETRY		REF
INFORMIXCONTIME		REF
INFORMIXCONCSMCFG	IDS 2000, IUS only	REF
INFORMIXDIR		REF
INFORMIXKEYTAB		REF
INFORMIXOPCACHE	OnLine/Optical only	REF

(2 of 4)

List of Environment Variables

Environment Variable	Restrictions	Reference
INFORMIXSERVER		REF
INFORMIXSHMBASE	IDS 2000, IDS 7.3, IUS, ODS, OWS only (UNIX)	REF
INFORMIXSQLHOSTS		REF
INFORMIXSTACKSIZE	IDS 2000, IDS 7.3, IUS, ODS, OWS only (all platforms)	REF
INFORMIXTERM	DB-Access only	REF
INF_ROL_SEP	IDS 2000, IUS, ODS only	REF
LANG	NLS only	REF
LC_COLLATE	NLS only	REF
LC_CTYPE	NLS only	REF
LC_MONETARY	NLS only	REF
LC_NUMERIC	NLS only	REF
LC_TIME	NLS only	REF
NODEFDAC		REF
ONCONFIG	IDS 2000, IDS 7.3, IUS, ODS, OWS only (all platforms)	REF
ONPLOAD	The High-Performance Loader only	REF, HPL
OPTCOMPIND	IDS 2000, IDS 7.3, IUS, ODS only (all platforms)	REF
OPTOFC	Informix ESQL/C Version 7.23 or later client applications	ESQL/C
PATH		REF
PDQPRIORITY	IDS 2000, IDS 7.3, IUS, ODS, XPS 8.3, AD/XP, XPS 8.11 only (all platforms)	REF
PLCONFIG	The High-Performance Loader only	REF

(3 of 4)

Environment Variable	Restrictions	Reference
PSORT_DBTEMP	IDS 2000, IDS 7.3, IUS, ODS only (all platforms)	REF
PSORT_NPROCS	IDS 2000, IDS 7.3, IUS, ODS, EDS, AD/XP, XPS 8.11 only (all platforms)	REF
RESTARTABLE_RESTORE		REF
SERVER_LOCALE	GLS only	GLS
SOURCE_REMOTE_SHELL	XPS 8.3 and AD/XP only	REF
SQLEXEC	ODS, SE only	REF
STMT_CACHE	IDS 2000 only	REF
STMT_CACHE_DEBUG	IDS 2000 only	REF
TERM	UNIX only	REF
TERMCAP	UNIX only	REF
TERMINFO	UNIX only	REF
THREADLIB	ESQL/C only	REF
XFER_CONFIG	XPS 8.3 and AD/XP only	REF

(4 of 4)

Upgrade and Reversion Messages

The dummy updates succeeded while converting database *database-name*.

Cause: During the upgrade of a database from a version earlier than Version 9.2, dummy update statements are run against the system tables in the database being upgraded. This message indicates successful completion of these updates.

Action: None required.

The dummy updates failed while upgrading database *database-name*.

Cause: During the upgrade of a database from a version earlier than Version 9.2, dummy update statements are run against the system tables in the database being upgraded. This message indicates failure in running one of these update statements.

Action: Based on the version from which you are upgrading, you need to run a particular dummy update script.

Converting database *database-name*.

Cause: This message appears at the start of the upgrade of each database in the system.

Action: None required.

The database *database-name* has been converted successfully.

Cause: This message indicates successful completion of an upgrade of the specified database.

Action: None required.

The conversion of the database *database-name* has failed.

Cause: Indicates that an upgrade of the specified database has failed.

Action: Connect to the database. This action triggers an upgrade of the database. If it fails, the relevant error message appears. Contact Informix Technical Support.

Must drop new database *database-name* before attempting reversion. Iserrno *error*.

Cause: The system contains a database that was created in a more recent version of the database server.

Action: Drop the new database and attempt reversion.

Cannot open **sysams** in database *database-name*, iserrno *error*.

Cause: An error occurred when the database server opened the **sysams** system table.

Action: Note the error number and contact Informix Technical Support.

Must drop newly created access methods in database *database-name*, iserrno *error*.

Cause: Some user-defined access methods are still present in the **sysams** system catalog table. The version to which you are reverting does not support access methods.

Action: Query the **sysams** system table and delete the problem entries and then attempt reversion.

Cannot open %s in database %s, iserrno %d.

Cause: An error occurred when the database server attempted to open the specified system table.

Action: Note the error number and contact Informix Technical Support.

Must drop records using new access methods from %s in database %s, iserrno %d.

Cause: The specified system table contains entries that use either system-defined or user-defined access methods, but the version to which you are reverting does not allow them.

Action: Query the relevant system table, delete the problem entries, and then attempt reversion.

Cannot open **systriggers** in database %s, iserrno %d.

Cause: An error occurred when the database server accessed the **systriggers** system table.

Action: Note the error number and contact Informix Technical Support.

Cannot open **systrigbody** in database %s, iserrno %d.

Cause: An error occurred when the database server accessed the **systrigbody** system table.

Action: Note the error number and contact Informix Technical Support.

Database %s: Must drop trigger (id = %d).

Cause: The database contains a trigger that was created in a version more recent than the one to which you are upgrading.

Action: Drop the trigger with the specified trigger identification number and then attempt reversion.

Cannot open **sysdistrib** in database %s, iserrno %d.

Cause: An error occurred when the database server accessed the **sysdistrib** system table.

Action: Note the error number and contact Informix Technical Support.

Must drop new user-defined statistics in database %s, iserrno %d.

Cause: Some distributions in the **sysdistrib** system table use user-defined statistics. This feature is not supported in the version to which you are reverting.

Action: Ensure that no user-defined statistics are present or used in the system and then attempt reversion.

Read failed. Table %s, Database %s, iserrno = %d.

Cause: An error occurred reading the specified system table.

Action: Note the error number and contact Informix Technical Support.

Must drop long identifiers in table %s in database %s.

Cause: Identifiers greater than 18 characters in length are not supported in the database server version to which you are reverting.

Action: Make sure that all long identifiers in the system are either dropped or renamed before you attempt reversion.

Memory allocation error.

Cause: The database server ran out of shared memory.

Action: Take one of the following actions:

1. Increase swap space on the computer.
2. Check kernel shared-memory parameters for limits on shared memory.
3. Decrease the size of the memory allocated with your **\$ONCONFIG** or **%ONCONFIG** settings such as **BUFFERS**.

Error in renaming **systdist**.

Cause: An internal error occurred while trying to find and rename the **Informix.systdist** SPL routine.

Action: Contact Informix Technical Support.

Error in slow altering a system table.

Cause: An internal error occurred while performing reversion.

Action: Contact Informix Technical Support.

Error in updating **sysindices**.

Cause: An internal error occurred while updating the **sysindices** table.

Action: Contact Informix Technical Support.

Error renaming procedures.

Cause: An internal error occurred while updating the **sysprocedures** table.

Action: Contact Informix Technical Support.

Error in dropping hash access method.

Cause: An internal error occurred while updating the **sysams** table.

Action: Contact Informix Technical Support.

Error in dropping system defined type (**Informix.stat**).

Cause: An internal error occurred during an update to the **sysxdtypes**, **sysctddesc**, or **sysxdttypeauth** system table.

Action: Contact Informix Technical Support.

Error in updating R-tree records in the **sysams** system table.

Cause: An internal error occurred while updating R-tree records in the **sysams** system table.

Action: Contact Informix Technical Support.

Cannot open **sysxdtypes** in database %s, iserrno %d.

Cause: An error occurred while accessing the **sysxdtypes** system table.

Action: Note the error number and contact Informix Technical Support.

Catalog reversion checks for database %s.

Cause: This message indicates the start, success, or failure of the database check phase during reversion.

Action: None required.

Starting reversion of database %s.

Cause: Indicates the start of actual reversion of the specified database.

Action: None required.

Reversion of database %s FAILED.

Cause: Indicates the failure of reversion of the specified database.

Action: None required.

Database %s SUCCESSFULLY reverted.

Cause: Indicates the success of reversion of the specified database.

Action: None required.

Checking database %s for revertibility.

Cause: Indicates that start of the reversion checks on the specified database.

Action: None required.

Database %s is revertible.

Cause: The database has passed all reversion checks and is revertible to the specified version.

Action: None required.

Database %s is not revertible.

Cause: The database has failed one of the reversion checks and is not revertible.

Action: Take action to correct the error displayed as a separate message.

Error in updating system-defined views.

Cause: An internal error occurred while updating the **systables** system table.

Action: Contact Informix Technical Support.

Error updating rows in **sysxdtypes**.

Cause: An internal error occurred while updating the **sysxdtypes** system table.

Action: Contact Informix Technical Support.

Cannot update **syscasts** in database %s. Iserrno %d.

Cause: An internal error occurred while inserting data into the **syscasts** system table.

Action: Contact Informix Technical Support.

Error in reverting R-tree index in database %s. Iserrno %d.

Cause: An error resulted during an R-tree index reversion.

Action: Note the error number and contact Informix Technical Support.

Internal server error.

Cause: An unexpected error occurred during database reversion.

Action: Contact Informix Technical Support.

Error removing **sysdistrib** row for tabid = %d, colid = %d in database %s. iserrno = %d.

Cause: An error occurred while updating the **sysdistrib** system table.

Action: Note the error number and contact Informix Technical Support.

Cannot revert constraint with id %d (in **syschecks**).

Cause: The database has a constraint that was defined in a version more recent than the one to which you are reverting.

Action: Drop the specified constraint and retry reversion.

Cannot revert new table fragment expression for table with id %d.

Cause: The fragmentation of this table was defined in a version more recent than the one to which you are reverting.

Action: Drop the problem table fragmentation scheme and retry reversion.

Cannot revert new fragment expression for index %s, tabid %d.

Cause: The index fragmentation was defined in a version more recent than the one to which you are reverting.

Action: Drop the problem index-fragmentation scheme and retry reversion.

R-tree error message conversion completed successfully.

Cause: The R-tree error message upgrade completed successfully.

Action: None required

R-tree error message conversion started.

Cause: The R-tree error message upgrade script is now running.

Action: None required.

R-tree error message conversion failed (See **/tmp/conR-tree.out**, **/tmp/conR-tree.databases**).

Cause: R-tree error message upgrade failed.

Action: See **/tmp/conR-tree.out** and **/tmp/R-tree.databases**.

WARNING: Target server version must have a certified storage manager installed after conversion/reversion and before bringing up server.

Cause: ON-Bar is being upgraded or reverted. You need to ensure that a storage manager is installed. The storage manager must be certified with the target database server version.

Action: None.

ON-Bar conversion start.

Cause: The ON-Bar upgrade script is now running.

Action: None.

ON-Bar conversion failed; see **/tmp/bar_conv.out**.

Cause: The ON-Bar upgrade failed.

Action: For failure details, see **/tmp/bar_conv.out**.

ON-Bar conversion completed successfully.

Cause: The ON-Bar upgrade completed successfully.

Action: None.

ON-Bar reversion start.

Cause: The ON-Bar reversion script is now running.

Action: None.

ON-Bar reversion failed; see **/tmp/bar_rev.out**.

Cause: ON-Bar reversion failed.

Action: For failure details, see **/tmp/bar_rev.out**.

ON-Bar reversion completed successfully.

Cause: ON-Bar reversion was completed successfully.

Action: None.

ON-Bar reversion test start.

Cause: ON-Bar reversion test script is now running.

Action: None.

ON-Bar reversion test completed successfully.

Cause: ON-Bar reversion test was completed successfully.

Action: None.

There is a semidetached index in this table, which cannot be reverted. Drop this index and retry reversion.

Cause: A semidetached index on this table cannot be reverted.

Action: To see the list of all semidetached indexes, refer to the database server message log. These indexes cannot be reverted. To continue reversion, drop these semidetached indexes and retry reversion. If needed, you will need to re-create these indexes after reversion is complete.

Converting '**onpload**' database.

Cause: This message appears in **online.log** at the beginning of an **onpload** upgrade.

Action: None required.

The '**onpload**' conversion completed successfully.

Cause: This message appears in **online.log** at the successful completion of an **onpload** conversion.

Action: None required.

The '**onpload**' conversion failed. For details, look in **\$INFORMIXDIR/etc/conpload.out**.

Cause: An upgrade of the **onpload** database failed.

Action: Find out the cause of failure from **\$INFORMIXDIR/etc/conpload.out**. Fix the problem before you reattempt the upgrade.

Reverting '**onpload**' database.

Cause: This message appears in **online.log** at the beginning of **onpload** reversion.

Action: None required.

The '**onpload**' reversion completed successfully.

Cause: Printed in **online.log** at the successful completion of reversion.

Action: None required.

The '**onpload**' reversion failed. For details, look in **\$INFORMIXDIR/etc/revpload.out**.

Cause: Reversion of the **onpload** database failed.

Action: Find the cause of failure in **\$INFORMIXDIR/etc/revpload.out**. Fix the problem before you reattempt reversion.

The '**onpload**' reversion test start.

Cause: Printed in **online.log** at the beginning of **onpload** reversion testing.

Action: None required.

The '**onpload**' reversion test completed successfully.

Cause: Printed in **online.log** if the **onpload** database is revertible.

Action: None required.

The **onpload** database contains load/unload jobs referring to long table names, column names, or database names. These jobs will not work as expected until they are redefined.

Cause: Printed during **onpload** reversion testing if the **onpload** database contains references to long table names, column names, or database names. But the reversion will complete.

Action: Redefine the load and unload jobs in the **onpload** database that have references to long identifiers.

Converting '**syscdr**' database.

Cause: This message appears in **online.log** at the beginning of an Enterprise Replication upgrade.

Action: None required.

The '**syscdr**' conversion completed successfully.

Cause: This message appears in **online.log** at the successful completion of a **syscdr** database upgrade.

Action: None required.

The '**syscdr**' conversion failed. For details, look in **\$INFORMIXDIR/etc/concdr.out**.

Cause: An upgrade of the **syscdr** database failed.

Action: Find the cause of the failure in **\$INFORMIXDIR/etc/concdr.out**. Fix the problem before you reattempt the upgrade.

Reverting '**syscdr**' database.

Cause: This message appears in **online.log** at the beginning of an Enterprise Replication upgrade.

Action: None required.

The '**syscdr**' reversion completed successfully.

Cause: Printed in **online.log** at the successful completion of the **syscdr** reversion.

Action: None required.

The '**syscdr**' reversion failed. For details, look in **\$INFORMIXDIR/etc/revcdr.out**.

Cause: Reversion of the **syscdr** database failed.

Action: Find the cause of failure in **\$INFORMIXDIR/etc/revcdr.out**. Fix the problem before you reattempt reversion.

CDR reversion test start.

Cause: Printed in **online.log** at the beginning of Enterprise Replication reversion testing.

Action: None required.

CDR reversion test completed successfully.

Cause: Printed in **online.log** if the **syscdr** database is revertible.

Action: None required.

CDR reversion test failed.

Cause: Printed in **online.log** if Enterprise Replication is not revertible.

Action: For more information, look at the messages prior to this message. Fix the reported problem before you attempt reversion.

Enterprise Replication should be in a stopped state for conversion/reversion to proceed.

Cause: Enterprise Replication is not in a stopped state.

Action: Stop Enterprise Replication before you proceed with an upgrade or reversion.

The Control and TRG send queues should be empty for conversion/reversion to proceed.

Cause: The Control and TRG send queues are not empty.

Action: Wait for these queues to drain before you attempt either an upgrade or reversion.

The 'syscdr' database contains references to long server names that are not supported in the earlier version(s) of the database server.

Cause: The Enterprise Replication global catalog contains long database server names.

Action: Redefine database servers with short names before you attempt reversion.

The dbspace names for send and receive queues or path names for ATS and RIS directories exceed pre-9.2 limits for this server.

Cause: The dbspace names for the send and receive queues for the database server being reverted are longer than 18 bytes. Or the path names for ATS and RIS directories are longer than 128 bytes.

Action: Modify the database server definition with shorter names before you attempt reversion.

The 'syscdr' database contains replicate definitions on long database, table, or owner names.

Cause: The global catalog has replicates defined on long table names, database names, or owner names from the database server being reverted.

Action: Drop the participants with long identifiers and add them again with shorter names.

The stored procedure specified for conflict resolution exceeds 18 bytes in length.

Cause: The SPL routine names specified for conflict resolution for some of the replicates for which this database server is a participant exceeds 18 bytes.

Action: Modify the replicate definitions with shorter SPL routine names.

The 'syscdr' database contains references to long replicate names.

Cause: The global catalog contains long replicate names.

Action: Drop and redefine replicates with long names before you attempt reversion.

The 'syscdr' database contains references to long group names.

Cause: The global catalog contains long group names.

Action: Drop and redefine groups with long names before you attempt reversion.

Index

A

- aaodir subdirectory 14-9, 14-16, 14-24, 16-6, 16-16
- Access methods
 - dropping indexes 6-25
 - retrieving external data 6-6
- Administration mode 11-37, 11-41, 14-25
- Administration tools 14-27
 - configuration 14-22
 - installing 14-12, 14-18, 16-12, 16-21
 - reverting 14-38
 - upgrading 14-27
 - Version 7.12 14-38
- adtcfg files 14-9, 14-16, 14-24, 16-6, 16-16
- adtmasks files 14-9, 14-16, 14-24, 16-6, 16-16
- AFF_NPROCS configuration
 - parameter 6-6
- AFF_SPROC configuration
 - parameter 6-6
- ALARMPROGRAM configuration
 - parameter 11-45, 14-41
- ALL keyword, with dbschema
 - utility 22-5
- All Servers tree view 14-35
- ALS client product 17-9, 17-15, 17-17, 17-18
- ALS database
 - supporting multibyte data 17-15
- ALS database server support for
 - GLS client 17-17, 17-19
- ALS locale
 - description of 17-15
 - saving in system catalog 17-15
- ALS. *See* Asian Language Support.
- ALS8BITFSYS environment
 - variable 17-16, 18-14
- ALTER FRAGMENT statement
 - 14-54
- ALTER TABLE statement 11-39, 14-39
- ANSI compliance level Intro-23
- ANSI database, moving 12-35
- ANSI logging 12-36
- Answers OnLine 11-22, 12-8, 14-24
- Applications, client
 - Communications Support
 - Module 4-19, 6-17
 - See also* Client applications.
- ARC_CONFIG file
 - saving 4-12, 6-11, 8-10, 10-25, 11-23
 - updating 4-20, 6-17
- Archive-management utility 12-22
- ASCII file, produced by unloading 2-8
- Asian Language Support (ALS)
 - ALS locale 17-15
 - client products 17-9, 17-15, 17-18
 - database servers 17-19
 - definition of 17-3, 17-14
 - description of 17-14
 - environment variables 17-16
 - GLS support for 17-17
 - locale 17-15
 - migrating from 4.x ASCII 18-19

- migrating from Version 5.x 18-15
- migrating from Version 6.x 18-13
- migrating to Version 4.x 18-17
- products 17-15
- storing database locale 17-15
- Audit configuration files 14-9, 14-16, 16-6, 16-16
- Authentication policy 4-19, 4-33, 6-16, 6-27

B

- Backing up
 - Dynamic Server 2000
 - after migration 4-25
 - before reversion 4-31
 - Dynamic Server 7.3
 - after migration 14-20, 16-23
 - before migration 4-16, 8-12, 11-25, 14-20
 - Dynamic Server with AD and XP
 - Options
 - after migration 9-12, 10-37
 - after reversion 7-26
 - Dynamic Server, Workgroup and Developer Editions
 - after migration 14-13, 16-14
 - before migration 14-13, 14-27
 - Extended Parallel Server
 - after migration 7-16, 8-17
 - before reversion 7-19
 - logical logs 4-21, 6-17
 - OnLine Dynamic Server
 - after migration 11-33
 - after reversion 11-54
 - before migration 4-16, 6-13, 8-12, 10-27, 11-37, 11-42
 - Version 5.x 12-27, 12-44
 - OnLine Workgroup Server 14-27
 - Version 7.12 14-45
 - Version 7.22 14-35
 - Universal Server
 - after migration 6-19
 - before migration 4-16
 - before reversion 6-22
- Backup and restore tool
 - Dynamic Server 7.3 14-20

- Dynamic Server, Workgroup and Developer Editions 14-13
- OnLine Workgroup Server 14-22
 - Version 7.12 14-27, 14-45
 - Version 7.22 14-35
- whole-system backup 14-13, 14-20, 16-14, 16-23
- bcheck utility, changed to secheck 13-5
- Binary files, loading 2-9, 2-13, 25-13
- BladeManager
 - installing and registering
 - DataBlade modules 4-20, 6-19
 - unregistering DataBlade modules 6-22
- blademgr utility 6-22
- bldutil.sh 9-11
- BLOB data type 6-24, 6-25
- Blobs, moving with onload 25-16
- Blobspace
 - adding 14-46
 - backing up 4-5, 6-4, 7-4, 8-4, 9-4, 10-4
 - disk space used 11-17
 - moving, with onunload and onload 25-16
 - removing 14-51
- Boldface type Intro-10
- BOOLEAN data type 6-24
- boot90.sql and boot901.sql 4-26, 6-20
- Bringing down
 - Dynamic Server 7.3 4-14, 4-17
 - Dynamic Server with AD and XP
 - Options 7-12
 - Dynamic Server, Workgroup and Developer Editions 16-20
- OnLine Dynamic Server 4-14, 4-17, 6-14, 10-28, 11-27, 12-28
- Online Workgroup Server 14-28
- Universal Server 4-14, 4-17
- Bringing up
 - Dynamic Server 2000 4-21
 - Dynamic Server 7.3 11-32, 14-19, 16-22
 - Dynamic Server with AD and XP
 - Options 10-32

- Dynamic Server, Workgroup and Developer Editions 14-13, 14-34, 16-13
- Extended Parallel Server 8-16
- OnLine Dynamic Server 4-34, 6-27, 12-32, 12-43
- OnLine Workgroup Server 14-29
- Universal Server 6-17
- B-tree index 6-25
- Buffer cache 12-7
- BUFFERS configuration parameter 12-41
- BYTE data type 14-46

C

- Calculating memory requirements for OnLine Dynamic Server 12-7
- Casts 6-24
- CC8BITLEVEL environment
 - variable 11-48, 14-42, 17-8, 17-16
- CDR_DSLOCKWAIT configuration parameter 11-13
- CDR_EVALTHREADS
 - configuration parameter 11-13
- CDR_LOGBUFFERS configuration parameter 11-13
- CDR_NIFMEMS configuration parameter 11-13
- CDR_NIFQUEUES configuration parameter 11-13
- CDR_NIFUSEHELP configuration parameter 11-13
- CDR_QUEUEMEM configuration parameter 11-13
- CHAR data type 18-12, 18-13, 19-8
- Character-position form of FILE and INSERT statements 21-15
- CHECK TABLE statement 14-50
- Checking available space
 - Dynamic Server 2000 4-10
 - Universal Server 6-9
- Checking database integrity 4-25, 5-4, 6-18, 7-11, 8-11, 10-36, 11-25, 11-42, 11-53, 12-43, 14-10, 14-16, 14-26, 16-6
- Choosing a migration method 1-3

CHUNKS configuration parameter 12-41

C-ISAM
 converting to SE 13-6
 migrating 1-11

Client applications
 ALS support 17-9, 17-15, 17-17, 17-18
 code-set conversion 17-8
 Communications Support Module 4-19, 6-17
 GLS support 17-6, 17-9
 NLS support 17-9, 17-11, 17-13

Client locale 17-8, 17-9
 See also Locale.

Client product
 ALS 17-9, 17-15, 17-18
 GLS 17-6, 17-9
 NLS 17-9, 17-11, 17-13

CLIENT_LOCALE environment
 variable 11-35, 11-48, 14-38
 ALS usage 17-9, 17-16
 default locale 17-6
 for NLS migration 18-10, 18-11
 GLS usage 17-8, 17-9
 set by database server 17-9

CLOB data type 6-24, 6-25

Code set
 ASCII 17-9
 converting 17-8
 multibyte 17-4, 17-14
 single-byte 17-4, 17-9

Code, sample, conventions for Intro-19

Code-set conversion
 HPL 2-20
 performing 17-8

Cogroups, creating 8-16, 10-32

COLLATION locale category 17-7, 17-11

COLLCHAR environment variable 11-35, 11-49, 14-38, 14-43, 17-12

Collection data type 6-24

Command Center
 installing 14-12, 14-18, 16-12, 16-21

Command file, dbload 21-9

Command-line conventions
 elements of Intro-17
 example diagram Intro-19
 how to read Intro-19

Command-line utilities 14-26

Commands
 export 12-30
 setenv 12-30
 See also SQL statement.

Comment icons Intro-11

COMMIT WORK statement 12-36

Communications Support Module
 client applications 4-19, 6-17
 configuring 4-19, 6-16
 removing 4-33, 6-27
 saving configuration 4-31, 6-21, 7-18

Compliance, with industry standards Intro-23

concsn.cfg file
 creating 4-19, 6-16
 deleting 4-33, 6-27
 saving 4-31, 6-21, 7-18

config.arc file, updating 4-20, 6-17

Configuration file
 customizing 11-29
 Dynamic Server 2000
 customizing 4-19, 4-33
 saving a copy 4-31

Dynamic Server 7.3
 customizing 11-26, 16-21
 saving a copy 14-9, 16-6

Dynamic Server 7.3, customizing 14-19

Dynamic Server with AD and XP
 Options, customizing 7-25, 10-31

Dynamic Server, Workgroup and Developer Editions
 customizing 14-12, 16-12
 saving a copy 14-16

Extended Parallel Server
 customizing 7-14, 8-15
 saving a copy 7-18

OnLine Dynamic Server
 saving a copy 10-24, 11-22, 12-25
 setting up, example 12-30

OnLine Workgroup Server,
 saving a copy 14-24

onxfer utility 26-4, 26-7

Universal Server
 customizing 6-16, 6-26
 saving a copy 6-21

Workgroup and Developer Editions, customizing 14-31

Configuration parameter
 AFF_NPROCS 6-6
 AFF_SPROC 6-6
 ALARMPROGRAM 11-45, 14-41
 BUFFERS 12-41
 CDR_DSLOCKWAIT 11-7
 CDR_EVALTHREADS 11-7
 CDR_LOGBUFFERS 11-7
 CDR_QUEUEMEM 11-7
 CHUNKS 12-41
 CONFIGSIZE 10-7, 10-8
 continuous data replication 11-13, 14-42
 COSERVER 10-8
 DBSERVERALIASES 12-30
 DBSERVERNAME 10-7, 10-8, 12-30
 DBSPACES 12-41
 DBSPACETEMP 12-15, 12-31
 DBTEMPSPACE 10-7, 10-8
 DIRECTIVES 11-7
 DS_ADM_POLICY 10-7, 10-8
 DS_MAX_SCANS 10-7, 10-8
 DS_TOTAL_MEMORY 10-7, 10-8
 Dynamic Server 7.3 12-20
 END 10-8
 FILLFACTOR 12-36
 HETERO_COMMIT 6-5, 11-45, 14-41
 ISM_DATA_POOL 11-7
 ISM_LOG_POOL 11-7
 ISO_CURLOCKS 10-7, 10-8
 LOCKS 12-21
 LOG_BACKUP_MODE 11-7
 LOGFILES 12-41
 LOGSMAX 12-41
 MAX_PDQPRIORITY 10-7, 10-8
 MIRRORPATH 10-7, 10-8
 MULTIPROCESSOR 6-6
 NOAGE 6-6
 NODE 10-8

NUMAIOVPS 6-6
 NUMCPUVPS 6-6
 ONLANGMAP 11-45, 14-41
 onxfer utility
 configuration file contents 26-7
 DEVICE 26-7
 FORMAT 26-9
 MSGPATH 26-10
 NSTREAMS 26-11
 REJECT_DIR 26-12
 SCHEMA_DIR 26-13
 SOURCE 26-14
 SOURCE_DIR 26-16
 TARGET 26-14
 TARGET_DIR 26-16
 OPT_GOAL 11-7
 OPTCOMPIND 11-16, 11-47
 PHYSSLICE 10-7, 10-8
 RESTARTABLE_RESTORE 11-7
 revised minimum values 12-20
 ROOTNAME 10-7, 10-8
 ROOTOFFSET 4-19, 6-16, 11-29,
 11-47, 14-12, 14-19, 14-32,
 16-12
 ROOTPATH 4-19, 6-16, 10-7,
 10-8, 11-29, 14-12, 14-19,
 14-32, 16-12
 ROOTSIZE 4-19, 6-16, 11-29,
 14-12, 14-19, 14-32, 16-12
 ROOTSLICE 10-7, 10-8
 SBSPACENAME 6-5, 6-26, 7-25
 SHMADD 12-9, 12-21
 SHMTOTAL 12-9, 12-22
 SHMVIRTSIZE 12-21
 SINGLE_CPU_VP 6-6
 SYSALARMPROGRAM 11-7
 TBLSPACE_stats 11-7
 VPCLASS 6-6, 6-26, 7-25
 configuration parameter
 FORMAT 26-9
 MSGPATH 26-10
 NSTREAMS 26-11
 REJECT_DIR 26-12
 SCHEMA_DIR 26-13
 SOURCE 26-14
 SOURCE_DIR 26-16
 TARGET 26-14
 TARGET_DIR 26-16

Configurations
 OnLine Workgroup Server 14-22
 Workgroup and Developer
 Editions 14-22
 Configuring
 open file descriptors 12-12
 semaphore parameters 12-10,
 12-24
 Contact information Intro-24
 Conventions, documentation
 Intro-9
 Converting. *See* Upgrading.
 Coserver name, description of 10-9
 Coserver number, part of coserver
 name 10-9
 CREATE AUDIT statement 14-50
 CREATE DATABASE statement
 14-7, 14-48, 14-52, 16-10, 16-20
 CREATE INDEX statement 12-19,
 12-36
 CREATE PROCEDURE statement,
 use of dbschema 22-6
 CREATE ROLE statement 14-54
 CREATE SYNONYM statement,
 use of dbschema 22-6
 CREATE TABLE statement
 loading data 14-7, 14-48, 14-52,
 16-10, 16-20
 removing information 14-51, 16-8
 use of dbschema 22-6
 CREATE VIEW statement, use of
 dbschema 22-6
 Creating a dbload command file
 21-9
 CSM. *See* Communications Support
 Module.
 CTYPE locale category 17-7, 17-11

D

Data distributions and disk
 utilization 12-13, 12-15
 Data type
 BOOLEAN 6-24
 BYTE 14-46
 CHAR 18-12, 18-13, 19-8
 CLOB and BLOB 6-24, 6-25
 collection 6-24
 DISTINCT 6-24

INT8 6-24
 LVARCHAR 6-24
 NCHAR 12-38, 17-20, 18-12,
 18-13, 19-6, 19-8
 NVARCHAR 12-38, 17-20, 18-12,
 18-13, 19-6, 19-8
 OPAQUE 6-24
 ROW 6-24
 SERIAL8 6-24
 TEXT 14-46
 VARCHAR 14-50, 18-12, 18-13,
 19-8
 Database
 English 17-4, 18-12
 non-English 17-4
 ownership, set by onload 25-15
 verifying integrity of 4-25, 5-4,
 6-18, 7-11, 8-11, 10-36, 11-24,
 11-42, 11-51, 11-53, 12-40,
 14-10, 14-16, 14-26, 16-6
 Database locale
 definition of 17-20
 determining current 18-4
 finding for ALS database 17-15
 finding for NLS database 18-6
 in ALS database 17-15
 in GLS database 17-7
 in NLS database 17-12
 set by database server 17-9
 specifying name of 17-8
 storing in database 17-20
 verifying 18-5
 See also Locale.
 Database server
 administration mode 11-37,
 11-41, 14-25
 ALS 17-19
 GLS 17-5
 NLS 17-10, 17-13
 platforms and versions 1-3, 1-11
 reinstalling 11-50, 14-43
 upgrading 14-27
 Database server definitions 14-37
 DataBlade modules
 data types 6-24
 deleting hidden tables 6-23
 hidden tables 6-23
 installing and registering 4-20,
 6-19
 unregistering 6-22

- DATAFILES 10-34
- Data-replication secondary server
 - guidelines 12-22
- DB_LOCALE environment variable
 - 11-35, 11-48, 14-38
 - ALS usage 17-9, 17-17
 - default locale 17-6
 - GLS usage 17-9
 - NLS usage 18-10, 18-11
 - set by database server 17-9
- DB-Access utility Intro-6
- dbaccess utility
 - creating a database 14-48, 14-52
 - estimating size of user table
 - indexes 12-17
 - updating statistics 12-14
 - using LOAD and UNLOAD
 - 14-47, 14-52
- DBAPICODE environment variable
 - 11-12, 17-9, 17-12
- DBCENTURY environment
 - variable 11-48, 14-42
- DBCODESET environment variable
 - 17-9, 17-16, 17-19, 18-15, 18-17
- DBCSOVERRIDE environment
 - variable 17-16
- DBDATE environment variable
 - 14-47, 17-12, 17-16
- dbexport and dbimport utilities
 - 12-6
- dbexport utility
 - c option 20-7
 - description of 2-14, 20-3
 - destination options 20-7
 - migrating to GLS support from
 - NLS 18-10
 - moving a database to
 - Dynamic Server 7.3 14-46
 - Dynamic Server, Workgroup and Developer Editions 14-46, 16-8
 - SE 14-51
 - schema output 20-10
 - ss option 14-46, 14-51, 20-7
- DBFLTMASK environment
 - variable 11-48, 14-42
- DBFORMAT environment variable
 - 17-16
- dbimport utility
 - c option 20-12
 - changing the locale 20-20
 - create options 20-15
 - database logging mode 20-17
 - description of 2-14
 - importing from another computer
 - 2-21
 - input file location options 20-13
 - Interrupt key 20-12
 - l options 20-17
 - migrating to GLS support from
 - NLS 18-10
 - moving a database to
 - Dynamic Server 7.3 14-47
 - Dynamic Server, Workgroup and Developer Editions 14-47, 16-8
 - SE 14-51
 - using with GLS 20-12
 - using with NLS 20-20
- DBINFXRC environment variable
 - 17-16
- DBLANG environment variable
 - 17-12, 17-16
- dbload utility
 - compared to LOAD 2-17
 - creating a command file 21-9
 - description of 21-3
 - e option 21-6
 - FILE statement 21-9
 - i option 21-6
 - importing from another computer
 - 2-21
 - INSERT statement
 - compared to SQL INSERT statement 21-17
 - using 21-9, 21-11
 - Interrupt key 21-7
 - moving data to
 - Dynamic Server 7.3 14-49, 15-6
 - Dynamic Server, Workgroup and Developer Editions 14-8, 14-49, 16-10
 - SE 14-53
 - number errors to allow 21-6
 - r option 21-6
 - speed, increasing 21-8
- syntax 21-3
- writing a command file
 - in character-position form 21-18
 - in delimiter form 21-13
- dblog utility, changed to selog 13-5
- DBMONEY environment variable
 - 14-47, 17-12, 17-17
- DB-Monitor utility 12-28
- DBNLS environment variable
 - 11-49, 14-43, 17-9, 17-12, 17-14
- DBONPLOAD environment
 - variable 11-48, 14-42
- DBPATH environment variable
 - and sort files 12-15
 - updating 14-54
- dbschema utility
 - create schema for a database 4-28, 6-21, 22-6
 - description of 2-17, 22-3
 - distribution information 22-18
 - Dynamic Server, Workgroup and Developer Editions
 - moving data 14-8, 16-10
 - moving selected tables 14-6, 16-9
 - moving data to
 - Dynamic Server 7.3 14-49, 15-6
 - Dynamic Server, Workgroup and Developer Editions 14-49
 - SE 14-53
 - moving selected tables to
 - Dynamic Server 7.3 14-47, 15-6
 - Dynamic Server, Workgroup and Developer Editions 14-47
 - SE 14-52
 - obtaining
 - privilege schema 22-12
 - synonym schema 22-11
 - owner conventions 22-7
 - saving database information 25-16
 - specifying a table, view, or procedure 22-14
 - specifying the role schema 22-17
 - ss option 22-8
- DBSERVERALIASES configuration
 - parameter 12-30

- dbservername
 - format and options 10-8
 - part of coserver name 10-9
- DBSERVERNAME configuration
 - parameter 12-30
- dbspaces, creating 8-16, 10-32
- Dbspace
 - adding 14-46
 - backing up 4-5, 6-4, 7-4, 8-4, 9-4, 10-4
 - defining disk space for
 - data distributions 12-15
 - upgrading 6-9, 12-12
 - user table indexes 12-17
 - defining temporary dbspaces 12-15
 - free pages 12-17
 - moving tables to another dbspace 25-16
 - removing 14-51
- DBSPACES configuration
 - parameter 12-41
- DBSPACETEMP configuration
 - parameter 12-15, 12-31
- DBSPACETEMP environment
 - variable 12-31
- dbssodir subdirectory 14-9, 14-16, 16-6, 16-16
- DBTIME environment variable
 - 17-13, 17-17
- Default locale Intro-6, 17-6, 18-12
- Delimiter form of FILE and INSERT statements 21-10
- Deluxe mode loading 10-20
- Demonstration databases Intro-6
- Dependencies, software Intro-5
- DEVICE configuration parameter 26-7
- Diagnostics table, displayed by
 - dbschema utility 2-18, 22-22
- Directory, installation 4-18, 6-15, 7-13, 8-14, 9-8, 10-29, 11-28, 12-29, 14-30
- Disk utilization
 - issues during upgrading 12-12
 - root dbspace 12-12
- Display schema for a database 22-6
- dist_pages formula 12-15
- DISTINCT data type 6-24
- Distribution information for tables 22-18
- Documentation notes Intro-22, 11-20, 12-4, 14-23
- Documentation, types of
 - documentation notes Intro-22
 - error message files Intro-21
 - machine notes Intro-22
 - on-line help Intro-21
 - on-line manuals Intro-20
 - printed manuals Intro-21
 - related reading Intro-23
 - release notes Intro-22
- DRDA database 2-22
- DROP AUDIT statement 14-50
- DROP INDEX SQL statement 12-19, 12-36
- DROP ROLE statement 14-54
- dropps.sql script 12-22
- Dynamic Server 2000
 - authentication policies 4-19, 4-33
 - backing up
 - after migration 4-25
 - before reversion 4-31
 - bringing up 4-21
 - DataBlade modules, installing 4-20
 - environments supported 1-4
 - GLS support 17-5
 - importing data 2-21
 - initializing 4-21
 - installing 4-17
 - language support 17-5
 - migration
 - between UNIX and Windows NT 1-11
 - summary 1-11
 - migration guidelines 4-20
 - migration procedure 4-20
 - moving data between computers 2-11, 2-13, 2-21
 - onload and onunload 2-12
 - release notes 4-4
 - reverting 4-26, 4-36
 - space requirements 4-10
 - starting 4-21
 - using dbexport and dbimport 2-13
 - using dbimport 2-21
 - using dbload 2-21
 - using onload and onunload 2-11, 2-12
- Dynamic Server 7.3
 - backing up
 - after migration 14-20, 16-23
 - before migration 4-16, 8-12, 11-25, 14-20
 - backup and restore tool 14-20
 - bringing down 4-17
 - bringing up 11-32, 14-19, 16-22
 - configuration issues 12-20
 - configuring
 - open file descriptors 12-12
 - semaphore parameters 12-10
 - environments supported 1-4
 - estimating
 - size of system catalog indexes 12-12
 - importing data 2-21
 - initiating fast recovery 4-14, 8-11
 - installing 11-27, 12-29, 14-18
 - migration
 - between UNIX and Windows NT 1-11
 - C-ISAM to IDS 7.3 14-49
 - summary 1-11
 - to Dynamic Server, Workgroup and Developer Editions 14-14, 16-15
 - to SE 1-11, 14-54
 - migration paths
 - moving data between computers 2-13, 2-21
 - release notes 11-19
 - SQL syntax restrictions 14-54
 - starting 11-32
 - upgrading user table indexes 12-16–12-19
 - using data distributions 12-15
 - using dbexport and dbimport 2-13
 - using dbimport 2-21
 - using dbload 2-21
 - verifying
 - data 11-42
 - database integrity 11-24
 - verifying database integrity 14-10

Dynamic Server with AD and XP Options

- backing up
 - after migration 9-12, 10-37
 - after reversion 7-26
- bringing down 7-12
- bringing up 10-32
- environments supported 1-4
- GLS support 10-18, 17-5
- initiating fast recovery 7-12
- language support 17-5
- release notes 7-13, 9-8, 10-4, 10-29
- reverting to 7-26
- starting 10-32

Dynamic Server with AD and XP Options, release notes 9-4, 9-8

Dynamic Server, Linux Edition, environments supported 1-4

Dynamic Server, Workgroup and Developer Editions

- backing up
 - after migration 14-13, 16-14
 - before migration 14-13, 14-27
- backup and restore tool 14-13, 16-14
- bringing up 14-13, 14-34, 16-13
- environments supported 1-4
- GLS support 17-5
- initiating fast recovery 14-26
- installing 14-12, 16-12
- migration
 - C-ISAM to WD 14-49
 - to SE 14-54
- SQL syntax restrictions 14-54
- starting 14-34
- verifying data integrity 14-26
- verifying database integrity 14-16, 16-6

E

EDA/SQL server. *See* Enterprise Gateway for EDA/SQL.

en_us.8859-1 locale Intro-6

See also U.S. English locale.

English database 17-4, 17-6, 18-7, 18-12

Extended Parallel Server

- backing up
 - after migration 7-16, 8-17
 - before reversion 7-19
- bringing up 8-16
- environments supported 1-4
- installing 8-14
- release notes 7-4, 7-13, 8-4, 8-15
- reversion restrictions 7-17
- reverting 8-18
- starting 8-16

Enterprise Gateway for EDA/SQL 2-22

Enterprise Gateway Manager 2-22

Enterprise Gateway with DRDA 2-22

Enterprise Replication

- configuration parameters 11-13
- configuring 11-30, 14-33
- stopping 11-39, 14-39

Environment variables

ALS8BITFSYS 17-16, 18-14

boldface type Intro-10

CC8BITLEVEL 11-48, 14-42, 17-8, 17-16

changes in OnLine Dynamic

Server 6.0 12-20

CLIENT_LOCALE 11-35, 11-48, 14-38, 17-8, 17-16

COLLCHAR 11-35, 11-49, 14-38, 14-43, 17-12

DB_LOCALE 11-35, 11-48, 14-38, 17-17

DBAPICODE 11-12, 17-9, 17-12

DBCENTURY 11-48, 14-42

DBCODASET 17-9, 17-16, 17-19

DBCSOEVERIDE 17-16

DBDATE 14-47, 17-12, 17-16

DBFLTMASK 11-48, 14-42

DBFORMAT 17-16

DBINFXRC 17-16

DBLANG 17-12, 17-16

DBMONEY 14-47, 17-12, 17-17

DBNLS 11-49, 14-43, 17-9, 17-12, 17-14

DBONPLOAD 11-48, 14-42

DBPATH 12-15, 14-54

DBSPACETEMP 12-15, 12-31

DBTIME 17-13, 17-17

definition of A-1

ESQLMF 11-48, 14-42, 17-8

for ALS support 17-16

for GLS support 17-8

for NLS support 17-12

GL_DATE 11-48, 14-42, 17-8

GL_DATETIME 11-48, 14-43, 17-8

GLS8BITFSYS 11-48, 14-42, 17-8, 18-14

ifx_directives 11-6

INFORMIXCONCSMCFG 6-5, 6-27

INFORMIXDIR 7-14, 8-13, 9-9, 10-30, 12-30

INFORMIXKEYTAB 6-5, 6-27, 11-6

INFORMIXSERVER 4-18, 6-16, 7-14, 8-13, 9-9, 10-5, 10-30, 12-20, 12-30, 14-47

INFORMIXSQLHOSTS 4-18, 6-16, 7-3, 7-7, 8-13, 10-30, 11-29, 14-31

LANG 11-49, 14-43, 17-9, 17-13, 17-14

LC_COLLATE 17-13

LC_CTYPE 17-13

LC_MONETARY 17-13

LC_NUMERIC 17-13

LC_TIME 17-13

list of A-4–A-7

LOG_BACKUP_MODE 11-9

new in Dynamic Server 2000 4-5

NODEFDAC 11-48

ONCONFIG 4-18, 6-16, 7-14, 8-13, 9-9, 10-30, 11-19, 12-20, 12-30

ONPLOAD 11-48, 14-43

OPTCOMPIND 11-16

PATH 4-18, 6-16, 7-14, 8-13, 9-9, 10-30, 12-30

PDQPRIORITY 10-5

PLCONFIG 11-48, 14-43

PSORT_NPROCS 10-5

reverting Dynamic Server 2000 4-33

reverting Dynamic Server with AD and XP Options 7-25

- reverting OnLine Dynamic Server
 - 11-48, 12-43
- reverting OnLine Workgroup Server
 - 14-42
- reverting Universal Server
 - 6-27
- SE, required
 - 13-4
- SERVER_LOCALE
 - 11-48, 14-43, 17-8
- SOURCE_REMOTE_SHELL
 - 26-4
- SQLEXEC
 - 12-20
- TBCONFIG
 - 11-19, 12-20
- THREADLIB
 - 11-48, 14-43
- types of A-2
- UNIX A-2
- Windows NT A-2
- XFER_CONFIG
 - 26-4
- environment variables
 - XFER_CONFIG
 - 26-4
- Error message files
 - Intro-21
- Error messages, DataBlade modules
 - 6-23
- ESQL/C
 - ALS support
 - 17-18
 - GLS support
 - 17-6
 - NLS support
 - 17-11
- ESQL/COBOL
 - GLS support
 - 17-6
 - NLS support
 - 17-11
- ESQLMF environment variable
 - 11-48, 14-42, 17-8
- Example
 - calculating free space
 - 4-11, 6-10, 8-9
 - DataBlade module cleanup
 - 6-23
 - onstat output
 - 4-16, 6-13, 7-10, 7-21, 8-12, 10-27
- export command
 - 12-30
- Express Mode Loading
 - 10-19
- Extended data types
 - 4-20, 6-19, 6-24
- Extents, checking
 - 4-25, 5-4, 6-18, 7-11, 8-11, 10-36, 11-25, 11-42, 11-53, 12-37, 14-10, 14-16, 14-26, 16-6
- External data, storage space
 - 6-6, 6-25
- External spaces
 - creating
 - 6-6
 - deleting
 - 6-25
 - dropping virtual tables
 - 6-25

- External tables
 - 10-33
- Extracting schema information
 - 14-6, 15-3, 16-4
- Extspace. *See* External spaces.

F

- Fast recovery, initiating
 - 4-14, 6-12, 7-12, 8-11, 10-25, 14-26
- Feature icons
 - Intro-11
- Features of this product, new
 - Intro-7
- File descriptors
 - 12-12
- FILE statement
 - character-position form
 - 21-15
 - delimiter form
 - 21-10
 - syntax for
 - character-position form
 - 21-16
 - delimiter form
 - 21-10
 - with dbload
 - 21-9
- Files
 - documentation notes
 - 11-20, 12-4, 14-23
 - editing OnLine Dynamic Server configuration
 - 12-30
 - oncfg
 - 7-9, 7-14, 7-25, 9-7
 - onconfig
 - Dynamic Server 2000
 - 4-19, 4-31
 - Dynamic Server 7.3
 - 14-9, 16-6
 - Dynamic Server with AD and XP Options
 - 10-31
 - Dynamic Server, Workgroup and Developer Editions
 - 14-16
 - Extended Parallel Server
 - 7-14, 7-18, 8-15
 - OnLine Dynamic Server
 - 10-24, 11-22, 12-25
 - OnLine Workgroup Server
 - 14-24
 - Universal Server
 - 6-16, 6-21
- onconfig.std
 - 10-24, 11-22, 12-25
- readme
 - 11-22, 12-8, 14-24
- release notes
 - 12-4, 14-23
- secure auditing, growth of
 - 12-20
- setting up onconfig
 - 12-30
- sqlhosts
 - 4-12, 4-18, 6-11, 6-16, 7-9, 7-14, 8-10, 8-13, 9-7, 9-9, 10-24, 10-30, 11-22, 12-31
- sqlhosts.demo
 - 12-31

- sysmaster_pre60
 - 12-22
- tbconfig
 - 12-25
- tbconfig.std
 - 12-25
- tctermcap
 - saving
 - 4-12, 6-11, 7-9, 8-10, 10-24, 11-23, 12-25
 - used by ON-Archive
 - 12-22
- termcap
 - 4-12, 6-11, 7-9, 8-10, 9-7, 11-22, 12-25
- xcfg
 - 7-9, 7-14, 7-25, 9-7
- FILLFACTOR configuration parameter
 - 12-36
- Find Error utility
 - Intro-22
- finderr utility
 - Intro-21
- FORMAT configuration parameter
 - 26-9
- Formula
 - dist_pages
 - 12-15
 - for growth_in_pages
 - 12-13
 - free space, Dynamic Server 2000
 - 4-11
 - free space, Universal Server
 - 6-10
 - idx_growth_pages
 - 12-16
- Fragmentation
 - adding
 - 14-46, 15-4, 15-5
 - removing
 - 14-51
- Free space required
 - Dynamic Server 2000
 - 4-11
 - Universal Server
 - 6-10
- Functional indexes, Universal Server
 - 6-25
- Functions, user-defined
 - 6-23

G

- Gateway, importing from another computer
 - 2-21
- Generic B-tree index
 - 6-25
- GL_DATE environment variable
 - 11-48, 14-42, 17-8
- GL_DATETIME environment variable
 - 11-48, 14-43, 17-8
- glfiles utility
 - 18-4
- Global Language Support (GLS)
 - Intro-6, 17-8
 - and dbimport utility
 - 20-12
 - client products
 - 17-6, 17-9
 - code-set conversion
 - 17-8
 - database servers
 - 17-5
 - definition of
 - 17-3, 17-4

Dynamic Server with AD and XP
Options 10-18
environment variables 17-8, A-2
finding locales 18-4
glfiles utility 18-4
GLS file location 18-5
GLS locale 17-6
locale 17-6
migrating databases 11-35, 14-38
operating system locale 18-5
products 17-5
reverting databases 11-44, 14-41
reverting from 19-3
reverting to ALS database 19-5
reverting to NLS database 14-41, 19-3
storing database locale 17-7
support for ALS databases 17-17
support for NLS databases 17-13
Universal Server 6-7
upgrading to 18-3
using onload and onunload 2-12, 25-13

GLS client product 17-6, 17-9

GLS database
definition of 17-6
reverting to ALS 19-5
reverting to NLS 19-3
storing database locale 17-7

GLS database server
definition of 17-5
opening NLS database 18-11
setting client locale 17-9
setting database locale 17-9

GLS locale
definition of 17-6
finding available 18-4
locale categories 17-7
localized collation order 18-12, 18-13
operating system 18-8
operating-system-independent 18-8, 18-9
reverting to ALS 19-5
reverting to NLS 19-3
saving in system catalog 17-7
types of 18-8
verifying 18-5

GLS. *See* Global Language Support.

GLS8BITFSYS environment
variable 11-48, 14-42, 17-8, 18-14

GRANT FRAGMENT statement
14-54

GRANT statement
role privileges 22-13
use of dbschema 22-6

Group name 11-30, 14-32, 14-33

growth_in_pages formula 12-13

H

Hardware requirements
Dynamic Server 7.3 11-21
OnLine Workgroup Server 14-24
Workgroup and Developer Editions 14-24

HETERO_COMMIT configuration
parameter 6-5, 11-45, 14-41

High Availability 11-10

High-Performance Loader (HPL)
loading ASCII or COBOL data 2-20

OnLine Workgroup Server, not supported 16-15

onpload utility 2-20
using 2-20

Hybrid Fragmentation 10-11

I

Icons
feature Intro-11
Important Intro-11
platform Intro-11
product Intro-11
syntax diagram Intro-14
Tip Intro-11
Warning Intro-11

IDS 7.3 on NT. *See* Dynamic Server 7.3.

IDS 7.3 on UNIX. *See* Dynamic Server 7.3.

IECC. *See* Informix Enterprise Command Center.

Important paragraphs, icon for Intro-11

Importing non-Informix data 2-21

Index
and dbload utility 21-6
bitmap 10-17
B-tree 6-25
checking 4-25, 5-4, 6-18, 7-11, 8-11, 10-36, 11-25, 11-42, 11-53, 12-43, 14-10, 14-16, 14-26, 16-6
rebuilding 4-21, 6-17
R-tree 6-25
Universal Server 6-23

Index space needed
idx_growth_pages formula 12-16
system catalog, disk utilization 12-13

Industry standards, compliance with Intro-23

Information Builders, Inc. *See* Gateway.

Informix Dynamic Server
GLS support 17-5
physical-memory requirement 12-7

Informix Enterprise Command Center 2-6, 10-17
configurations 14-22
starting Dynamic Server 7.3 16-22
starting Dynamic Server, Workgroup and Developer Editions 16-13

Informix Enterprise Gateway. *See* Gateway.

Informix language support. *See* Language support.

Informix Storage Manager 4-8, 7-7, 8-6, 9-5, 10-18, 11-21, 12-23

informix user password 14-37

INFORMIXCONSCMCFG
environment variable 6-5, 6-27

INFORMIXDIR environment
variable 12-30
location of locale files 18-5

INFORMIXDIR/bin directory
Intro-7

INFORMIX-ESQL/C. *See* ESQL/C.

INFORMIX-ESQL/COBOL. *See* ESQL/COBOL.

INFORMIXKEYTAB environment variable 6-5, 6-27, 11-6
 INFORMIXSERVER environment variable 7-14, 8-13, 9-9, 10-30
 changing 14-47
 Dynamic Server 2000 4-18
 OnLine Dynamic Server 12-20, 12-30
 Universal Server 6-16
 INFORMIXSQLHOSTS environment variable
 Dynamic Server 2000 4-18
 Dynamic Server 7.3 11-29
 Universal Server 6-16
 Workgroup and Developer Editions 14-31
 Initial- and next-extent values 14-46, 14-51
 Initializing, Dynamic Server 2000 4-21
 In-Place ALTER TABLE 11-37, 11-40
 INSERT SQL statement 12-34
 INSERT statement
 character-position form 21-15
 delimiter form 21-10
 syntax for
 character-position form 21-16
 delimiter form 21-10
 with dbload 21-9
 inssnmp.exe utility 11-31, 14-30, 14-33
 Installation directory 4-18, 6-15, 7-13, 8-14, 9-8, 10-29, 11-28, 12-29, 14-30
 Installing
 Command Center 14-12, 14-18, 16-12, 16-21
 Dynamic Server 2000 4-17
 Dynamic Server 7.3 11-27, 12-29, 14-18
 Dynamic Server with AD and XP Options
 release notes 9-8, 10-29
 Dynamic Server, Workgroup and Developer Editions 14-12, 16-12

 Extended Parallel Server, release notes 7-13, 8-14
 OnLine Dynamic Server 12-29
 storage manager 4-8, 7-7, 8-6, 9-5, 10-18, 11-21, 12-23
 Universal Server 6-15
 Workgroup and Developer Editions 14-29
 INT8 data type 6-24
 Interrupt key
 with dbimport 20-12
 with dbload 21-7
 Interruptions, preventing 12-40
 ISM. *See* Informix Storage Manager.
 ISO 8859-1 code set Intro-6
 isql utility 12-17

L

LANG environment variable 11-49, 14-43, 17-9, 17-13, 17-14
 Language support
 ALS 17-14
 GLS 17-4, 17-13, 17-17
 in database servers 17-20
 NLS 17-9
 types of 17-3
 understanding 17-4
 LC_COLLATE environment variable 17-13
 LC_CTYPE environment variable 17-13
 LC_MONETARY environment variable 17-13
 LC_NUMERIC environment variable 17-13
 LC_TIME environment variable 17-13
 Level-0 backup 14-27
 after moving data 25-16
 Dynamic Server 2000 4-25
 Dynamic Server 7.3 4-16, 8-12
 Dynamic Server with AD and XP Options 7-11
 OnLine Dynamic Server 4-16, 4-36, 6-14, 8-12, 10-27, 12-26
 Universal Server 4-16, 6-19
 LIST data type 6-24

LOAD SQL statement
 converting to non-default locales 12-38
 moving selected tables to
 Dynamic Server 7.3 14-48, 16-20
 Dynamic Server, Workgroup and Developer Editions 14-7, 14-48, 16-10
 SE 14-53
 restriction 12-6
 syntax 23-5
 Loading
 ASCII files 2-8, 2-21
 binary data 2-9, 2-13, 25-13
 COBOL data 2-8, 2-20
 Locale Intro-6
 ALS 17-15
 definition of 17-4
 file location 18-5
 finding information about 18-4
 GLS 17-6
 GLS locale categories 17-7
 NLS 17-11
 NLS locale categories 17-11
 reverting to ALS from GLS 19-5
 reverting to NLS from GLS 19-3
 Lock mode for tables 14-46, 14-51
 Locking, set by onload 25-15
 LOCKS configuration parameter 11-17, 12-21
 LOGFILES configuration parameter 12-41
 Logging, and
 ANSI compliance 12-36
 onload and onunload 25-16
 Logical logs
 backing up 4-21, 6-17, 11-32, 12-32, 14-34
 changing mode 14-51
 logging status 16-8, 16-18
 minimum page total in OnLine Dynamic Server 12-21
 out of space 4-21, 6-17
 restoring 4-25, 6-19, 7-8, 7-16, 7-18, 9-6, 10-37, 11-33, 12-37, 14-36

- space required for Dynamic Server 2000 4-10
- space required for Universal Server 6-9
- usage status 12-21
- LOGSMAX configuration parameter 12-41
- Long transaction, risk when upgrading indexes 12-36
- LTAPEDEV parameter with onunload/onload 25-16
- LVARCHAR data type 6-24

M

- Machine notes Intro-22
- makeps.sql script 12-22
- Message file for error messages Intro-21
- Message segment, estimating size of 12-8
- MESSAGES locale category 17-7
- Messages, upgrade and reversion B-1
- Middle Eastern language support 11-35, 14-38
- Migrating a database to different operating system 1-11
 - Dynamic Server 7.3
 - using dbexport and dbimport 14-46, 15-6
 - using UNLOAD, dbschema, and LOAD 14-47, 15-6
 - Dynamic Server, Workgroup and Developer Editions
 - using dbexport and dbimport 14-46, 16-8
 - using UNLOAD, dbschema, and LOAD 14-6, 14-47, 16-9
- overview 2-3
- same platforms 2-9
- SE
 - using dbexport and dbimport 14-50
 - using UNLOAD, dbschema, and LOAD 14-52
- Migration
 - onload utility 25-4, 25-9
 - onunload utility 25-4
 - paths 1-3

- Migration procedure
 - between UNIX and Windows NT 15-5
 - between versions of Dynamic Server 7.3 11-34
 - between versions of OnLine Dynamic Server 11-5, 11-34, 12-38
 - between versions of SE 13-4
 - Dynamic Server 7.3 to Dynamic Server 2000 4-20
 - Dynamic Server 7.3 to Dynamic Server, Workgroup and Developer Editions 14-8, 16-10
 - Dynamic Server 7.3 to SE 14-50–14-54
 - Dynamic Server, Workgroup and Developer Editions to SE 14-50–14-54
 - GLS to ALS language support 19-5
 - GLS to NLS language support 19-3
 - OnLine 4.1 or 5.0 to Dynamic Server 7.3 12-24
 - OnLine 4.1 or 5.0 to OnLine Dynamic Server 7.2 12-24
 - OnLine Dynamic Server to Dynamic Server 2000 4-20
 - OnLine Dynamic Server to Universal Server 6-19
 - SE to Dynamic Server 7.3 14-45–14-49
 - SE to Dynamic Server, Workgroup and Developer Editions 14-45–14-49
 - Universal Server to Dynamic Server 2000 4-20
- Mirroring 12-15
- MONETARY locale category 17-7, 17-11
- Moving data
 - blobspaces 25-16
 - choosing a migration method 1-3
 - C-ISAM, migrating 13-6
 - constraints 2-7
 - oncheck utility 25-16
 - steps for using onunload and onload 25-16

- Moving to another operating system, migration paths 1-11
- MSGPATH configuration parameter 26-10
- Multibyte character 17-4, 17-14
- Multiple residency, definition of 11-9
- MULTIPROCESSOR configuration parameter 6-6
- MULTISET data type 6-24

N

- Native Language Support (NLS)
 - client products 17-9, 17-11, 17-13
 - database servers 17-10, 17-13
 - definition of 17-3, 17-9
 - description of 17-9
 - determining compatible GLS locales 18-8
 - environment variables 17-12, A-2
 - finding the database locale 18-6
 - GLS support for 12-38, 17-13
 - locale 17-11
 - NLS locale 17-11
 - populating with dbimport 20-20
 - products 17-10
 - setting NLS locales 14-41
 - setting NLS support 11-44, 14-41
 - storing database locale 17-12
- NCHAR data type 12-38, 17-20, 18-12, 18-13, 19-6, 19-8
- Network File System 10-21
- New features of this product Intro-7
- NFILE, NFILES, NOFILE, NOFILES OS parameters 12-12
- NFS. *See* Network File System.
- NLS client product 17-9, 17-11, 17-13
- NLS database
 - definition of 17-11
 - English 18-7, 18-12
 - non-English 18-8, 18-11
 - opening 18-11
 - saving database locale 17-12
 - unloading 18-10
- NLS database server
 - definition of 17-10
 - support for GLS client 17-13

NLS locale
 definition of 17-11
 description of 17-11
 locale categories 17-11
 saving in system catalog 17-12, 18-6

NLS. *See* Native Language Support.

NOAGE configuration parameter 6-6

NODEFDAC environment variable 11-48

Non-English database
 ALS support 17-15, 19-8
 definition of 17-4
 migration of 17-20
 NLS support 17-10, 18-8, 18-11
 onmode utility 19-5

Non-Informix data, importing 2-21

NSTREAMS configuration
 parameter 26-11

NUMAIOVPS configuration
 parameter 6-6

NUMCPUVPS configuration
 parameter 6-6

NUMERIC locale category 17-7, 17-11

NVARCHAR data type 12-38, 17-20, 18-12, 18-13, 19-6, 19-8

O

Off-line mode, verifying 12-28

ON-Archive
 backing up
 Dynamic Server, Workgroup and Developer Editions 11-9
 OnLine Workgroup Server 11-18

ON-Archive utility
 and ttermcap file 12-22
 and the sysmaster database 12-14
 and Windows NT 11-25, 11-42, 14-10, 14-17, 16-7, 16-14, 16-17, 16-23
 backing up 14-35
 Dynamic Server 2000 4-31
 Dynamic Server 7.3 8-12

OnLine Dynamic Server 4-17, 6-14, 8-12, 10-27, 11-33, 12-37
 Universal Server 6-19, 6-22
 updating configuration files 4-20, 6-17

ON-Bar utility 14-35
 backing up
 Dynamic Server 2000 4-25, 4-31
 Dynamic Server 7.3 11-33
 Dynamic Server with AD and XP Options 7-16, 7-26, 9-12, 10-37
 Extended Parallel Server 7-19, 8-17
 Universal Server 6-19, 6-22

oncfg file in \$INFORMIXDIR/etc 7-9, 7-14, 7-25, 9-7

oncheck utility
 before moving data 25-16
 changed from tbcheck 12-6
 rebuilding table indexes 4-21, 6-17
 upgrading indexes 12-18, 12-35, 12-36
 verifying database integrity 4-25, 6-18, 10-36, 11-24, 11-42, 11-51, 11-53, 12-40, 12-43, 14-26

ONCONFIG environment variable
 Dynamic Server 2000 4-18
 Dynamic Server 7.3 11-19
 OnLine Dynamic Server 12-20, 12-30
 Universal Server 6-16

onconfig file
 customizing 11-29, 14-12, 14-19, 14-31, 16-12
 saving
 Dynamic Server 2000 4-31
 Dynamic Server 7.3 14-9, 16-6
 Dynamic Server, Workgroup and Developer Editions 14-16
 Extended Parallel Server 7-18
 OnLine Dynamic Server 10-24, 11-22, 12-25

OnLine Workgroup Server 14-24
 Universal Server 6-21
 updating 11-45, 14-41

ONCONFIG file. *See* Configuration file.

ONCONFIG parameters. *See* Configuration parameter.

onconfig.std file 10-24, 11-22, 12-25

oninit
 starting
 Dynamic Server with AD and XP Options 9-8
 OnLine Workgroup Server 11-32

oninit utility
 Dynamic Server 7.3 14-19
 Dynamic Server, Workgroup and Developer Editions for UNIX 14-13, 16-13
 -s option 4-14, 6-12, 7-12, 8-11, 9-8, 10-25, 11-32, 14-34
 starting
 Dynamic Server 2000 4-21
 Dynamic Server 7.3 11-32
 Extended Parallel Server 8-16
 OnLine Dynamic Server 11-32, 12-32
 Universal Server 6-17
 starting Dynamic Server, Workgroup and Developer Editions 14-34

ONLANGMAP configuration
 parameter 11-45, 14-41

OnLine
 environments supported 1-5
 reverting to 4.x ALS 19-8
 reverting to 5.x ALS 19-7

OnLine Dynamic Server
 ALS support 17-15, 17-21
 backing up
 after migration 11-33
 after reversion 11-54
 before migration 4-16, 6-13, 8-12, 10-27, 11-37, 11-42
 Version 5.x 12-27, 12-44
 bringing down 4-17, 6-14, 10-28, 11-27, 12-28

- bringing up 4-34, 6-27, 12-32, 12-43
- configuring semaphore parameters 12-10, 12-24
- copying configuration files 12-25
- environment variables A-4–A-7
- environments supported 1-4
- estimating memory requirements 12-7
- GLS support 17-5
- initiating fast recovery 4-14, 6-12, 8-11, 10-25
- installing 12-29
- language support 17-5, 17-10, 17-21, 17-22
- migration
 - between ALS and GLS support 17-23
 - between NLS and GLS support 17-23
 - between UNIX and Windows NT 15-5
 - to Dynamic Server 7.3 1-11
 - to OnLine Workgroup Server 1-11
 - to SE 1-11, 14-50
 - to UNIX 1-11
 - to Workgroup and Developer Editions 1-11
- NLS support 17-10
- obtaining database locale 18-7
- preventing interruption of users 12-40
- reinstalling 11-50
- reverting to
 - Version 4.x, 5.x 12-39–12-44
 - Version 6.0 or later 11-36–11-53, 19-4
 - Version 6.x ALS 19-6
 - Version 7.1x NLS 19-4
- revise configuration parameter values 12-20
- shutting down 12-40
- starting 4-34, 6-27, 12-32, 12-43
- UNIX versions 11-36
- upgrading to 7.2x 17-22
- upgrading user table indexes 12-33
- verifying

- database integrity 11-42, 11-51, 11-53, 12-40
- off-line mode 12-28
- warning 11-50
- On-line help Intro-21
- On-line manuals Intro-20
- OnLine Workgroup Server
 - backing up 14-27
 - Version 7.12 14-45
 - Version 7.22 14-35
 - backup and restore tool 14-27
 - bringing down 16-20
 - environment variables A-4–A-7
 - environments supported 1-5
 - GLS support 17-5
 - language support 17-5, 17-10, 17-21, 17-24
 - migration
 - between NLS and GLS support 17-24
 - to SE 14-50
 - moving data between computers 2-13, 2-21
 - NLS support 17-10
 - obtaining database locale 18-7
 - PDQ support 14-14, 16-15
 - release notes 14-23
 - shutting down 16-20
 - upgrading to 7.2x 17-24
 - using onload and onunload 2-13, 2-21
 - verifying database integrity 14-26
 - versions 14-22
 - warning 14-44
- OnLine XPS
 - environments supported 1-4
 - GLS support 17-5
 - language support 17-21
 - obtaining database locale 18-7
 - release notes 9-8
- onload and onunload utilities 2-11, 12-6, 12-34, 25-16
- onload utility
 - constraints on use 2-12, 25-13
 - create options 25-10, 25-11
 - logging status 25-14
 - moving a database 25-16
 - moving a table 25-16
 - moving locales 2-12, 25-13

- moving to another dbspace 25-16
- ownership and privileges 25-15
- specifying tape parameters 25-10
- steps for using 25-16
- syntax 25-10
- using between computers 2-9, 2-11
- onmode utility
 - and shared-memory connections 12-10
 - b 5.0 option 19-7
 - b 6.0 option 19-4
 - b 6.0A option 19-6
 - b 7.1 option 19-4
 - b 7.1UD1 option 19-4
 - b option 14-40, 24-4
 - changing mode to quiescent 12-35
 - ky option 4-14, 6-12, 7-12, 8-10, 10-25
 - m option 4-36, 6-28, 7-26, 11-32, 12-33, 14-34
 - quiescent to on-line mode 11-32, 12-33, 14-34
 - reversion syntax 19-3
 - reverting to 5.x ALS 19-7
 - reverting to 6.x ALS 19-6
 - reverting to 6.x NLS 19-4
 - reverting to 7.1x NLS 19-4
 - reverting to NLS database server 11-43, 19-3, 19-4
 - reverting to OnLine 5.0 12-41
 - reverting to OnLine Workgroup Server 7.12 14-40
 - reverting to Version 7.2 6-26
 - s option 12-35
 - shutting down 16-20
 - Dynamic Server 7.3 4-14, 8-10
 - Dynamic Server with AD and XP Options 7-12
 - OnLine Dynamic Server 4-14, 6-12, 8-10, 10-25
 - Universal Server 4-14
 - sy option 4-14, 6-12, 7-12, 8-10, 10-25, 11-23, 11-41, 14-11, 14-17, 14-25, 16-11
 - treatment of systables 19-5
- ONPLOAD environment variable 11-48, 14-43
- onpload utility 2-20

onsnmp utility, configuring 11-31,
 14-30, 14-33
 onspaces -d command 6-25
 onstat utility 4-16, 6-13, 7-10, 7-13,
 7-21, 8-12, 10-27, 11-24, 11-41,
 14-11, 14-18, 14-25, 16-11
 ontape utility
 backing up
 Dynamic Server 2000 4-25, 4-31
 Dynamic Server 7.3 8-12
 Dynamic Server, Workgroup
 and Developer Editions
 14-35
 logical logs 11-32, 12-32, 14-34
 OnLine Dynamic Server 4-16,
 4-36, 6-14, 8-12, 10-27,
 11-33, 12-37
 Universal Server 6-19, 6-22
 onunload and onload restriction
 12-6
 onunload utility
 constraints on use 2-12, 25-7,
 25-13
 locking 25-9
 logging mode 25-8
 moving a database 25-16
 moving a table 25-16
 moving locales 2-12, 25-13
 moving to another dbspace 25-16
 ownership and privileges 25-7
 steps for using 25-16
 syntax 25-5
 system catalog tables 25-8
 use without tape parameters 25-5
 using 2-9, 2-11
 what is included with a
 database 25-8
 table 25-8
 onutil 10-15
 onxfer utility 26-9, 26-10, 26-11,
 26-12, 26-13, 26-14, 26-16
 command syntax 26-5
 configuration file
 default name 26-4, 26-7
 description 26-7
 configuration parameter
 DEVICE 26-7
 configuration parameters 26-7

data format 26-9
 data streams 26-11
 description 26-3
 destination device 26-7
 environment variables 26-4
 guidelines 26-17
 message log 26-10
 moving data 26-3
 reject file location 26-12
 restrictions 26-17
 schema file location 26-13
 source data 26-14
 source database server 26-16
 SOURCE_REMOTE_SHELL
 environment variable 26-4
 target database or tables 26-14
 target database server 26-16
 XFER_CONFIG file 26-4
 OPAQUE data type 6-24
 oper_deflt.arc file
 saving 4-12, 6-11, 8-10, 10-25,
 11-23
 updating 4-20, 6-17
 Operating system
 reconfiguring 4-17, 6-15, 8-13,
 10-29, 11-27, 12-31, 14-29
 updates 12-12
 Operating-system locale. *See* NLS
 locale.
 Operating-system parameter
 NFILE 12-12
 NFILES 12-12
 NOFILE 12-12
 NOFILES 12-12
 SEMMNI 12-10
 SEMMNS 12-11
 SEMMSL 12-11
 SHMMAX 12-9
 SHMMNI 12-9
 SHMSEG 12-10
 SHMSIZE 12-9
 Operating-system resources
 disk utilization 12-12
 file descriptors 12-12
 secure auditing log files 12-20
 semaphores 12-10
 shared memory 12-7–12-10

Operating-system-independent
 locale. *See* GLS locale.
 Operational table, description of
 10-14
 Operators 6-24
 OPTCOMPIND configuration
 parameter 11-16, 11-47
 OPTCOMPIND environment
 variable 11-16
 options field 6-6, 10-9
 osahosts 14-24, 14-29
 Owner, in dbschema 22-7

P

Parallel data query (PDQ) 14-14,
 16-15
 PATH environment variable
 Dynamic Server 2000 4-18
 OnLine Dynamic Server 12-30
 Universal Server 6-16
 PDQ. *See* Parallel data query.
 Performance comparison 4-25,
 6-19, 12-23
 Physical memory requirement for
 Informix Dynamic Server 12-7
 Planning
 data migration overview 2-3
 migrating to Workgroup and
 Developer Editions 14-23
 upgrading OnLine Dynamic
 Server 12-3–12-23
 upgrading to Dynamic Server 7.3
 11-19
 Platform icons Intro-11
 Platforms, moving data between
 compatible computers 2-9, 2-13,
 2-21
 operating systems 1-11
 PLCONFIG environment variable
 11-48, 14-43
 Preparations for migrating to
 OnLine Dynamic Server 6.0 12-3
 Printed manuals Intro-21
 Privileges
 access privileges for tables 25-8
 required for onload 25-15
 required for onunload 25-7

Product icons Intro-11

Program group

Documentation notes Intro-23

Release notes Intro-23

Pseudotables, in sysmaster
database 12-14

R

Raw table, description of 10-14

read_ods.txt file 11-22, 12-8

Readme file 11-22, 12-8, 14-24

Rebuild system catalog indexes
12-33

Reconfiguring the operating system
4-17, 6-15, 8-13, 10-29, 11-27,
12-31, 14-29

RECOVER TABLE statement 14-50

Registering DataBlade modules
4-20, 6-19

Registry keys

SNMP master agent 11-31, 14-30,
14-33

sqlhosts 14-29

Reinstalling OnLine Dynamic
Server 11-50

REJECT_DIR configuration
parameter 26-12

REJECTFILE 10-37

Related reading Intro-23

Release notes Intro-22, 12-4

Dynamic Server 2000 4-4

Dynamic Server with AD and XP
Options 7-13, 9-4, 9-8, 10-4,
10-29

Extended Parallel Server 7-4,
7-13, 8-4, 8-15

Online Workgroup Server 14-23

OnLine XPS 9-8

Universal Server 6-4

Workgroup and Developer
Editions 14-23

RENAME DATABASE statement
14-54

REPAIR TABLE statement 14-50

Replication Manager 11-30, 14-32,
14-33

Requirements, system 11-21, 14-24

Reserve pages, checking 4-25, 5-4,
6-18, 7-11, 8-11, 10-36, 11-25,
11-42, 11-53, 12-43, 14-10,
14-16, 14-26, 16-6

Resident segment, estimating size
of 12-8

Restoring

data 14-27

OnLine Dynamic Server data
11-52, 11-54

OnLine Workgroup Server data
14-45

Reversion messages B-1

Reverting

administration tools 14-38

Dynamic Server 2000 4-26, 4-36

Dynamic Server with AD and XP
Options 7-26

Extended Parallel Server 8-18

Enterprise Replication 11-39,
14-39

GLS to ALS database 19-5

GLS to NLS database 19-3

modifying configuration
parameters 11-45, 14-41

OnLine Dynamic Server to
6.0 or later 11-36–11-52, 24-4
pre-6.0 12-39–12-44, 24-4

reinstalling the product 11-50

resetting environment variables
11-48

Universal Server 6-20, 6-28

REVOKE FRAGMENT statement
14-54

rofferr utility Intro-21

ROLLBACK WORK statement
12-36

ROLLFORWARD DATABASE
statement 14-50

Root dbspace

disk utilization 12-12

free pages 12-14

ROOTOFFSET configuration
parameter 4-19, 6-16, 11-29,
11-47, 14-12, 14-19, 14-32,
16-12

ROOTPATH configuration
parameter 4-19, 6-16, 11-29,
14-12, 14-19, 14-32, 16-12

ROOTSIZE configuration

parameter 4-19, 6-16, 11-29,
14-12, 14-19, 14-32, 16-12

Routines, user-defined 6-23

ROW data type 6-24

RSAM shared-memory segment
12-7

R-tree indexes 6-25

S

sales_demo database Intro-6

Sample-code conventions Intro-19

SBS PACENAME configuration

parameter 6-5, 6-26, 7-25

Sbspaces

about 6-6

deleting 6-25

SCHEMA_DIR configuration

parameter 26-13

SE

ALS support 17-15, 17-21

converting C-ISAM files 13-6

converting unsupported data
types 14-50

environment variables A-4–A-7

environments supported 1-5

GLS support 17-5

language support 17-5, 17-10,
17-21, 17-23

migration to

Dynamic Server 7.3 1-11,
14-45–14-49

Dynamic Server, Workgroup
and Developer Editions
14-45–14-49

GLS support 17-23

OnLine Dynamic Server 1-11

OnLine Workgroup Server 1-11

UNIX 1-11

Windows NT 1-11

Workgroup and Developer
Editions 1-11

NLS support 17-10

obtaining database locale 18-7

onload and onunload 2-12

reverting to 4.x ALS 19-8

reverting to 5.x ALS 19-7

reverting to 6.x ALS 19-6

- reverting to 6.x NLS 19-5
- reverting to 7.1x NLS 19-5
- reverting to 7.2x 18-11
- reverting to an earlier version 13-6
- SQL syntax restrictions 14-50
- steps for migration 13-5
- upgrading to 7.2x 17-23
- version levels 1-5
- Secure auditing, growth of log files 12-20
- Segment
 - message 12-7
 - resident 12-7
 - virtual 12-7
- SELECT statement 12-17, 12-19
- Semaphores 12-10
- SEMMNI OS parameter 12-10
- SEMMNS OS parameter 12-11
- SEMMSL OS parameter 12-11
- Serial columns 10-12
- SERIAL data type, treatment by dbschema 22-6
- Serial number and key 14-29
- SERIAL8 data type 6-24
- Server menu 14-35
- SERVER_LOCALE environment variable 11-48, 14-43, 17-8
- Services file 14-12, 14-19, 16-12, 16-21
- Services, Windows NT 14-11, 14-17, 16-11
- SET CONSTRAINTS statement 14-54
- SET data type 6-24
- SET DATASKIP statement 14-54
- SET ISOLATION statement 14-54
- SET LOG statement 14-54
- SET PDQPRIORITY statement 14-54
- SET ROLE statement 14-54
- SET SESSION AUTHORIZATION statement 14-54
- setenv command 12-30
- setenv.cmd 10-16
- Setnet32 tool 11-29, 14-31
- Setup tool 14-37
- Shared server machine 14-37
- Shared-memory communication interface 12-10
- Shared-memory segments
 - added at run time 12-9
 - estimating size of 12-7–12-10
 - in the operating system 12-9
- SHMADD configuration parameter 12-9, 12-21
- SHMMAX OS parameter 12-9
- SHMMNI OS parameter 12-9
- SHMSEG OS parameter 12-10
- SHMSIZE OS parameter 12-9
- SHMTOTAL configuration parameter 12-9, 12-22
- SHMVIRTSIZE configuration parameter 12-8, 12-21
- Shutting down
 - Dynamic Server, Workgroup and Developer Editions 16-20
 - Enterprise Replication 11-39, 14-39
 - OnLine Dynamic Server 12-40
- Simple large objects, moving with onload 25-16
- SINGLE_CPU_VP configuration parameter 6-6
- Single-byte character 17-4, 17-9
- Smart large objects
 - creating storage space 6-6
 - data types 6-24
 - deleting storage space 6-25
 - removing 6-24
- smi_load utility 11-38
- SNMP, configuring 11-31, 14-30, 14-33
- Software dependencies Intro-5
- Sort files and DBSPACETEMP 12-15
- SOURCE configuration parameter 26-14
- SOURCE_DIR configuration parameter 26-16
- SOURCE_REMOTE_SHELL environment variable 26-4
- Space
 - additional pages for system catalog indexes 12-13
 - for sysmaster database 4-10, 6-9, 12-14
- Space requirements
 - Dynamic Server 7.3 11-21
 - Workgroup and Developer Editions
 - OnLine Workgroup Server 14-24
- SPL routines, Universal Server 6-25
- SQL code Intro-19
- SQL reserved words 4-6, 11-7
- SQL statement
 - ALTER FRAGMENT 14-54
 - CHECK TABLE 14-50
 - COMMIT WORK 12-36
 - CREATE AUDIT 14-50
 - CREATE DATABASE 14-7, 14-48, 14-52, 16-10, 16-20
 - CREATE INDEX 12-19, 12-36
 - CREATE ROLE 14-54
 - CREATE TABLE
 - loading data 14-7, 14-48, 14-52, 16-10, 16-20
 - removing information 14-51, 16-8
 - DROP AUDIT 14-50
 - DROP INDEX 12-19, 12-36
 - DROP ROLE 14-54
 - FILE 21-9
 - GRANT FRAGMENT 14-54
 - INSERT 12-34, 21-9
 - LOAD 12-6, 12-38
 - RECOVER TABLE 14-50
 - remove unsupported syntax
 - 14-39, 14-48, 16-8, 16-18
 - RENAME DATABASE 14-54
 - REPAIR TABLE 14-50
 - REVOKE FRAGMENT 14-54
 - ROLLBACK WORK 12-36
 - ROLLFORWARD DATABASE 14-50
 - SELECT 12-17, 12-19
 - SET DATASKIP 14-54
 - SET LOG 14-54
 - SET PDQPRIORITY 14-54
 - SET ROLE 14-54
 - SET SESSION AUTHORIZATION 14-54
 - START DATABASE 14-50

syntax recognized by
 Dynamic Server 7.3 14-54
 Dynamic Server, Workgroup
 and Developer Editions
 14-54
 SE 14-50
 UNLOAD 12-6, 12-38
 UPDATE STATISTICS
 data distributions 12-15, 22-18
 queries 11-33, 14-14, 14-20,
 14-36, 16-14, 16-23
 system catalog indexes 12-14
 user table indexes 12-18, 12-35
 SQLEXEC environment variable
 12-20
 sqlhosts file, UNIX
 changing name or path 4-18,
 6-16, 7-14, 8-13, 9-9, 10-30
 csm option 4-19, 4-33, 6-16, 6-27
 modify, example 12-30, 12-31
 pre-6.0 OnLine Dynamic Server
 13-4
 save a copy when migrating 4-12,
 6-11, 7-9, 8-10, 9-7, 10-24,
 11-22, 12-25
 sqlhosts, Windows NT
 database server definitions 14-12,
 14-18, 16-12, 16-21
 registry 14-29
 sqlhosts.demo file 12-31
 sqlturbo processes 12-21
 srv_agent2 16-21
 Standard table, description of 10-14
 START DATABASE statement
 14-50
 Starting
 Dynamic Server 2000 4-21
 Dynamic Server 7.3 11-32, 14-19,
 16-22
 Dynamic Server with AD and XP
 Options 10-32
 Dynamic Server, Workgroup and
 Developer Editions 14-13,
 14-34, 16-13
 Extended Parallel Server 8-16
 OnLine Dynamic Server 4-34,
 6-27, 12-32, 12-43
 OnLine Workgroup Server 6-26
 Universal Server 6-17

Static table, description of 10-14
 Storage manager 4-8, 7-7, 8-6, 9-5,
 10-18, 11-21, 12-23
 stores_demo database Intro-6
 superstores_demo database Intro-7
 Swap space, size of 12-7
 Syntax conventions
 description of Intro-13
 example diagram Intro-16
 icons used in Intro-14
 Syntax diagrams, elements in
 Intro-14
 sysbldiprvided table 6-23
 sysbldirequired table 6-23
 sysbldobjdepends table 6-23
 sysbldobjects table 6-23
 sysbldregistered table 6-23
 sysdbslocale table 18-5
 syserrors catalog, UGEN errors 6-23
 sysmaster database
 and disk utilization 12-13
 and logical logs 4-21, 6-17, 12-21
 automatic creation of 12-32
 pre-existing 12-22
 space required to build 4-10, 6-9,
 11-21, 12-14
 sysdbslocale table 18-5
 systabnames table 18-7
 sysmaster_pre60 file 12-22
 systables table
 ALS support 19-5
 ALS usage 17-15, 18-4, 18-13
 finding database locale 18-7
 GLS usage 17-7
 NLS usage 17-12, 18-4, 18-6,
 18-12, 19-3
 storing database locale 17-20
 systabnames table 18-7
 System catalog indexes
 disk utilization 12-13
 estimating pages needed 12-12
 rebuild 12-33
 System catalogs
 boot scripts, Dynamic Server 2000
 4-26
 boot scripts, Universal Server 6-20

 checking tables 4-25, 5-4, 6-18,
 7-11, 8-11, 10-36, 11-25,
 11-42, 11-53, 12-43, 14-10,
 14-16, 14-26, 16-6
 size of, Universal Server 6-18
 storing database locale 17-20
 System requirements
 database Intro-5
 software Intro-5
 System resources, additional
 required for 6.0 12-6

T

Table type
 logging 10-13
 nonlogging 10-13
 operational 10-13
 permanent 10-13
 raw 10-13
 scratch 10-13
 standard 10-13
 static 10-13
 temp 10-13
 temporary 10-13
 Tables
 estimate additional index pages
 12-17
 finding location of 12-17
 in separate dbspaces, disk
 utilization 12-17
 moving to another dbspace 12-34
 TAPEDEV parameter
 with onunload and onload 25-16
 TARGET configuration parameter
 26-14
 TARGET_DIR configuration
 parameter 26-16
 tbcheck utility
 replaced by oncheck 11-19, 12-6
 verifying the integrity of data
 12-27
 TBCONFIG environment variable
 11-19, 12-20
 tbconfig file 12-25
 tbconfig.std file 12-25
 tbinit utility 12-26, 12-43
 tblog and tbunload utilities 12-6

tbmode utility 12-26, 12-28
 tbmonitor utility 12-28
 tbstat utility
 - option 12-27
 -d option 12-14, 12-17, 12-25
 -l option 12-21
 tbtape utility 12-6
 tctermcap file
 saving 4-12, 6-11, 7-9, 8-10,
 10-24, 11-23, 12-25
 used by ON-Archive 12-22
 Temporary files and tables 12-15
 termcap file in \$INFORMIXDIR/etc
 4-12, 6-11, 7-9, 8-10, 9-7, 11-22,
 12-25
 TEXT data type 14-46
 THREADLIB environment variable
 11-48, 14-43
 TIME locale category 17-7, 17-11
 Tip icons Intro-11
 Tools, migration 2-8

U

U.S. English locale 17-6, 18-17,
 18-19, 19-7
 Universal Server
 authentication policies 4-33, 6-16,
 6-27
 backing up
 after migration 6-19
 before migration 4-16
 before reversion 6-22
 bringing down 4-17
 bringing up 6-17
 casts 6-24
 DataBlade modules, installing
 6-19
 dropping indexes 6-23, 6-25
 environments supported 1-4
 extended data types 6-24
 GLS support 6-7, 17-5
 initiating fast recovery 4-14
 installing 6-15
 language support 17-5, 17-21
 migration guidelines 6-19
 migration procedure 6-19
 onload and onunload 2-12

operators 6-24
 release notes 6-4
 reverting 6-20, 6-28
 space requirements 6-9, 6-18
 SPL routines 6-25
 starting 6-17
 system catalogs 6-18
 user-defined data types 6-24
 user-defined routines 6-23
 using onload and onunload 2-12
 UNIX
 environment variables A-2
 migration paths 1-11
 OnLine Dynamic Server versions
 11-36
 SE versions 1-5
 starting Dynamic Server 7.3 14-19
 starting Dynamic Server,
 Workgroup and Developer
 Editions 14-13, 16-13
 UNLOAD SQL statement
 moving data 14-8, 16-10, 16-20
 moving data to
 Dynamic Server 7.3 14-49, 15-6
 Dynamic Server, Workgroup
 and Developer Editions
 14-49
 SE 14-53
 moving selected tables to
 Dynamic Server 7.3 14-47
 Dynamic Server, Workgroup
 and Developer Editions
 14-7, 14-47, 16-9
 SE 14-52
 restriction 12-6
 syntax 2-15, 23-4
 using nondefault locales 12-38
 Unregistering DataBlade modules
 6-22
 UPDATE STATISTICS statement
 data distributions 12-15, 22-18
 queries 11-33, 14-14, 14-20,
 14-36, 16-14
 system catalog indexes 12-14
 user table indexes 12-18, 12-35
 Updating ON-Archive
 configuration 4-20, 6-17
 Upgrade messages B-1

Upgrade screen 14-29
 Upgrading
 from OnLine 5.0 12-38
 from OnLine Dynamic Server 6.0
 or later 11-5, 11-34
 ISM 4-8, 7-7, 8-6, 9-5, 10-18,
 11-21, 12-23
 scenarios
 Dynamic Server, Workgroup
 and Developer Editions
 14-22
 OnLine Workgroup Server
 14-22
 to Dynamic Server 2000 4-20
 to Universal Server 6-19
 User-defined functions 6-23
 User-defined index, Universal
 Server 6-25
 User-defined routines 6-23
 USERS configuration parameter
 12-8
 Users, types of Intro-4
 User-table indexes
 and disk utilization 12-13
 preparing for upgrading to 6.0 or
 later 12-18
 risk of long transaction when
 upgrading 12-36
 upgrading in OnLine Dynamic
 Server 12-33
 Utilities
 changes, OnLine Dynamic Server
 6.0 12-6
 dbaccess 12-14, 12-17
 dbexport 2-14, 12-6, 20-3
 dbimport 2-14, 12-6
 dbload 21-3
 DB-Monitor 12-28
 dbschema 2-17, 22-3
 dropps.sql script 12-22
 gfiles 18-4
 isql 12-17
 makeps.sql script 12-22
 ON-Archive. *See* ON-Archive
 utility.
 ON-Bar. *See* ON-Bar utility.
 oncheck. *See* oncheck utility.
 oninit. *See* oninit utility.

- onload and onunload 2-11, 12-6, 12-34, 25-16
- onmode. *See* onmode utility.
- onpload 2-20
- onspaces 6-25
- onstat 4-16, 6-13, 7-10, 7-13, 7-21, 8-12, 10-27
- ontape. *See* ontape utility.
- onxfer. *See* onxfer utility.
- smi_load 11-38
- tbcheck 12-6, 12-27
- tbinit 12-26, 12-43
- tbload and tbunload 12-6
- tbmode 12-26, 12-28
- tbmonitor 12-28
- tbstat. *See* tbstat utility.
- tbtape 12-6

V

- Validated storage manager 4-8, 7-7, 8-6, 9-5, 10-18, 11-21, 12-23
- VARCHAR data type 14-50, 18-12, 18-13, 19-8
- Verifying database integrity 4-25, 6-18, 10-36, 12-37, 14-26
- Verifying storage-manager validation 4-8, 7-7, 8-6, 9-5, 10-18, 11-21, 12-23
- Versions, Informix database servers 1-3, 1-11
- View, display description with dbschema 22-6
- Virtual processors 12-10
- Virtual segment
 - estimating the size of 12-8
 - use of 12-7, 12-21
- Virtual tables 6-25
- VPCLASS configuration parameter 6-6, 6-26, 7-25

W

- Warning icons Intro-11
- Whole-system backup 14-35
- Windows NT
 - environment variables A-2
 - migration paths 1-11
 - SE versions 1-5

- Workgroup and Developer Editions
 - installing 14-27, 14-29
 - release notes 14-23
 - upgrading 14-27
 - versions 14-22
- Workgroup Replication
 - configuration parameters 11-13, 14-42
 - configuring 11-31, 14-30, 14-33
- Writing a dbload command file
 - in character-position form 21-18
 - in delimiter form 21-13

X

- X/Open compliance level Intro-23
- xcfg file in \$INFORMIXDIR/etc
 - 7-9, 7-14, 7-25, 9-7
- xctl 10-15
- XFER_CONFIG environment
 - variable 26-4
- xfer_config file 26-7

Symbols

- \$ARC_CONFIG. *See* ARC_CONFIG file.

